P802.1Qdy - RSTP/MSTP YANG – Reusability across SDOs (version 07)
Murugan Balraj (Nokia)
(murugan.balraj@nokia.com)
Introduction

This presentation

• Highlights a problem in the proposed YANG model for RSTP, MSTP as part of P802.1Qdy that would prevent its reuse in other SDOs like BBF.

• Presents a possible solution that would allow the YANG model to be reused by other SDOs, but at the same time preserve the overall structure and functionality of the YANG from IEEE 802.1 perspective.
The problem (RSTP)

48.6 YANG modules

Insert 48.6.26 after 48.6.25 (inserted by IEEE Std 802.1Qdx-2024) as follows:

48.6.26 The ieee802-dot1q-rstp YANG module

module ieee802-dot1q-rstp {
  yang-version 1.1;
  namespace "urn:ieee:std:802.1Q:yang:ieee802-dot1q-rstp";
  prefix rstp;

  
  augment "/dot1q:bridges/dot1q:bridge/dot1q:component"
    description "Augment Bridge with RSTP configuration.";
    container rstp {
      presence "The presence of this container indicates that RSTP is supported";
    }

  augment "/if:interfaces/if:interface/dot1q:bridge-port"
    description "Augment Bridge Port with RSTP configuration";
    reference "13.24, 13.25, and 13.27 of IEEE Std 802.1Q.";
    container rstp {
      presence "The presence of this container indicates that RSTP is supported";
    }

The “augments” clause present in the YANG module cannot be overridden and this makes it impossible for other SDOs to use this YANG module since they may not have the notion of Bridge / Component
The problem (MSTP)

The "augments" clause present in the YANG module cannot be overridden and this makes it impossible for other SDOs to use this YANG module since they may not have the notion of Bridge / Component.
BBF access devices are primarily aggregation devices that are multiplexing traffic in the upstream direction from multiple end users / customer ports towards the core network and in the downstream direction, demultiplexing traffic from the core network towards the end users / customer ports.

When a customer port is used purely to provide end user broadband connectivity, there would be no need for resiliency protocols like ERPS, RSTP/MSTP, etc.

There may be some cases where a customer port, mostly an ethernet port is used as an Uplink / Network port.

(The port type / usage is configurable in BBF). For such ports, resiliency protocols like ERPS, RSTP/MSTP would become applicable.

Network interfaces of BBF layer-2 access node are IEEE 802.1 & IEEE 802.3 compliant. It can interop with other layer-2 IEEE 802.1 devices (non-BBF) in a network topology.

Multiple network ports could also be bundled into a LAG interface and connected to a peer device in the core network.

When more than 1 network port / LAG interface is connected to the same network segment (Broadcast domain), then the topology could be a **Ring, Mesh or even a Daisy Chain** and resiliency protocols like ERPS or RSTP/MSTP would be required to avoid loops.

Each of the customer / end user side ports provide last mile connectivity to Residential or Business users and also e.g. serve as a backhaul connectivity for Mobile base stations. Various Layer 2 Access technologies are used including xDSL, PON and also Ethernet.

On the customer premises, the xDSL, PON or ethernet lines are terminated by CPEs/ONTs. The customer ports on the BBF access devices are logically considered to be different network segments and layer 2 switching / forwarding is generally not done across different customer ports (local / User 2 User Forwarding) on the same Access Device.

The BBF access devices could have one or more network ports connecting to the core network and each of these network ports could be connected to the same or different network segment (Broadcast domain).
BBF Layer-2 Access device and IEEE 802.1 Provider Edge Bridge

BBF Layer-2 Access device (forwarding)
- In a BBF Layer2 Access device, the SVLAN component and CVLAN components are not logically modelled separately but they have a concept of “forwarder/forwarder-ports/VLAN-sub-interface” and all the SVLAN and CVLAN aware “forwarder-ports” could be part of the same “forwarder”.
- VID configuration, traffic classification and the traffic manipulation rules are modelled on top of a BBF specific logical interface called as “VLAN-sub-interface”.
- Multiple “VLAN-sub-interface” for example using different VID configuration can be created on top of an IEEE 802.3 physical interface or IEEE 802.1AX LAG interface.
- “forwarder” responsible to route traffic across different “forwarder-ports” which has one to one association with the “VLAN-sub-interface”. For example, traffic received with VLAN 100 from a user port, route to the network port with VLAN 200.
- “forwarder” has a view of list VIDs which are needed for the “MST tree to list of VID” mapping.

Reference:
IEEE 802.1Q-2022, Section “15.4 C-tagged service interface”, Figure 15-6 – Customer Edge Ports (CEP)
BBF Layer-2 Access device YANG view (hardware/component/interface/L2-forwarding)

BBF specific YANG modules augments and/or leafref the IETF hardware/component, IETF system and IETF interfaces YANG modules. 

```
/hw:hardware/hw:component
/sys:system
/if:interfaces-state/if:interface
```

IEEE 802.1 YANG hierarchy
```
"/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-vlan/dot1q:vid-to-fid",
"/dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:filtering-database/dot1q:filtering-entry" etc.
```

BBF specific interface-type on top of IETF interface-type to configure VLAN tag match criteria, other configurations and state data

```
bbf-sub-interface-tagging.yang
```

E.g. `ifType ethernetCsMaCd, ieee8023adLag...`

```
interface(*) name (key) ...
```

```
vlan-sub-interface(*) name (key) ...
match-criteria
inner-vlan-tag
outer-vlan-tag ....
```

```
forwarder(*) name (key)
forwarding-database
forwarder-ports list(*) ...
```

```
bbf-l2-forwarding-forwarders.yang
bbf-l2-forwarding-forwarding-databases.yang
```

```
bbf-sub-interface-tagging.yang
```

```
0...n
```

```
0...n
```

```
0...n
```

```
0...n
```

```
0...n
```

```
0...n
```

```
0...n
```

```
0...n
```

```
0...n
```

```
0...n
```
BBF layer-2 access device – Operation of Spanning Tree Protocol Entity (BPDU receive/transmit possible view)

Spanning Tree Protocol Entity

LLC

Port State

ISS

Port Transmit and Receive

LLC

Port State

ISS

Port Transmit and Receive

802.n (ethernet1)

Ref : IEEE Std 802.1Q 2022, Figure 8-6

802.n (ethernet2)

BBF devices would need the port level RSTP/MSTP parameters to be configured on top of IETF interfaces.

BBF layer-2 access device capable of generating/receiving IEEE 802.1Q compliant RSTP/MSTP BPDUs.
Eighty-twos-n (ethernet1) 

VSI-1 [VLAN:100] 
VSI-2 [VLAN:200] 
VSI-3 [VLAN:100] 
VSI-4 [VLAN:200] 

Filtering Database 

Each forwarder has a list of forwarder-ports. Implicitly list VSIs(VIDs) for the MST trees.

VSI: 
• Match criteria - VID
• Traffic classification rules (single/dual/prio/untagged)
• Traffic manipulation actions (push tag, pop tag, rewrite)

Spanning Tree Protocol Entity

Ref: IEEE Std 802.1Q 2022, Figure 8-6
BBF layer-2 access device – Relaying MAC Frames

802.n (ethernet1)

- Frame Reception
- EISS
- LLC MS
- Port State Information
- VSI-1 [VLAN:100]
- fw-port1
- VSI: • Match criteria • Frame classification rules (single/dual/prio/untagged) • Frame manipulation actions (push tag, pop tag, rewrite)

forwarder2

- Forwarding Process (forwarder1)
- Filtering Database
- MST config information
- Port State Information

fw-port2

VSI-2 [VLAN:100]

802.n (ethernet2)

- Frame Transmit
- LLC MS
- Port State Information

Ref: IEEE Std 802.1s 2002, Figure 8-4
Proven strategy for reusability across SDOs
The YANG module that other SDOs like BBF would use in their devices has to be free of “augment” clauses and references to Bridge / Component / Bridge Port.

• Split the YANG module into two modules. The Base YANG module would contain the protocol specific YANG definitions and an IEEE 802.1 specific module would augment this base module and contain the “augment” clauses.

• Other SDOs like BBF could then use just the Base YANG module.

Following are some examples of existing IEEE 802.1 YANG models that are reusable

(Path: https://github.com/YangModels/yang/blob/main/standard/ieee/published/802.1)

<table>
<thead>
<tr>
<th>Domain</th>
<th>Common reusable YANG</th>
<th>IEEE 802.1 Bridge specific YANG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>These base modules are reusable in other SDOs</td>
<td>These IEEE 802.1 specific modules augment the base module by adding the “augment” and/or leafref clauses that link the base module to Bridge/Component/Bridge Port</td>
</tr>
<tr>
<td>CFM</td>
<td>ieee802-dot1q-cfm.yang</td>
<td>ieee802-dot1q-cfm-bridge.yang</td>
</tr>
<tr>
<td>Scheduler</td>
<td>ieee802-dot1q-sched.yang</td>
<td>ieee802-dot1q-sched-bridge.yang</td>
</tr>
<tr>
<td>PSFP</td>
<td>ieee802-dot1q-psfp.yang</td>
<td>ieee802-dot1q-psfp-bridge.yang</td>
</tr>
<tr>
<td>Stream Filters and Gates</td>
<td>ieee802-dot1q-stream-filters-gates.yang</td>
<td>ieee802-dot1q-stream-filters-gates-bridge.yang</td>
</tr>
<tr>
<td>Preemption</td>
<td>ieee802-dot1q-preemption.yang</td>
<td>ieee802-dot1q-preemption-bridge.yang</td>
</tr>
<tr>
<td>Congestion-isolation</td>
<td>ieee802-dot1q-congestion-isolation.yang</td>
<td>ieee802-dot1q-congestion-isolation-bridge.yang</td>
</tr>
</tbody>
</table>
Example of some existing IEEE 802.1 YANG models

### IEEE 802.1 Base CFM model

**(Protocol specific YANG definitions)**

<table>
<thead>
<tr>
<th>augment</th>
<th>iee802-dot1q-cfm.yang</th>
</tr>
</thead>
</table>

---

### IEEE 802.1 Bridge specific CFM augmentation

<table>
<thead>
<tr>
<th>augment</th>
<th>iee802-dot1q-cfm-bridge.yang</th>
</tr>
</thead>
</table>

- augment "/dot1q-cfm:cfm/dot1q-cfm:maintenance-group"
  --- bridge-name -> leafref to IEEE 802.1 Bridge
  --- bridge-component-name -> leafref to IEEE 802.1 Bridge component

- augment "/dot1q-cfm:cfm/dot1q-cfm:maintenance-group/dot1q-cfm:mep"
  --- interface-ref -> leafref to IETF interface

---

### BBF specific CFM augmentation

| augment | bbf-dot1q-cfm-interfaces.yang  
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bbf-dot1q-cfm-l2-forwarding.yang</td>
</tr>
</tbody>
</table>

- augment "/dot1q-cfm:cfm/dot1q-cfm:maintenance-group"
  --- forwarder -> leafref to BBF Forwarder

- augment "/dot1q-cfm:cfm/dot1q-cfm:maintenance-group/dot1q-cfm:mep"
  --- interface-ref -> leafref to BBF vlan-sub-interface

Note: Recommendation ITU-T Y.1731 CFM also uses the IEEE 802.1 Base CFM model to augment ITU-T Y.1731 specific CFM YANG definitions.

---

Ref: Recommendation ITU-T G.8052

---
Example of some existing IEEE 802.1 YANG Models

The IEEE 802.1 Bridge specific YANG module has a “uses” statement that augments the “Component” with the “grouping” from the base module.

PSFP common reusable YANG definitions

```
iedee802-dot1q-psfp.yang
```

IEEE 802.1 Bridge specific augmentation

```
iedee802-dot1q-psfp-bridge.yang

augment "/dot1q:bridges/dot1q:bridge/dot1q:component {
  uses psfp:psfp-parameters
}
```

Other SDO specific augmentation

Other SDOs like BBF could also utilize the “uses” statement to include the “grouping”.

Other YANG models “Scheduler, Stream Filters and Gates, Preemption and Congestion-isolation” also follow the same approach.

The Base module has a “grouping” statement and the SDO specific module achieves the augmentation with the “uses” statement.

Does not have references to Bridge / Component / Bridge Port

The base YANG module uses the “grouping” statement
P802.1Qdy decoupling proposal – overall strategy

Implementation in a device conforming to IEEE 802.1 specifications

Device conforming to IEEE 802.1 specifications would implement the YANG model of:

1. Protocol specific aspects YANG definition (Reusable across SDOs)
2. IEEE 802.1 device specific YANG augmentation
Device conforming to BBF specifications would implement the YANG model of:

1. Protocol specific aspects YANG definition (reusable across SDOs)
2. BBF device specific YANG augmentation
3. BBF Device implementing RSTP/MSTP Protocol
IEEE 802.1Qdy decoupling proposal – more details

IEEE 802.1Qdy current YANG definition

IEEE 802.1 device specific YANG module which uses Base YANG module

Per component MSTP augment

YANG module "ieee802-dot1q-mstp-bridge.yang" :

- augment "/dotq:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-mst" :
  - uses mstp-per-component-protocol-specific-data

Per interface MSTP augment

YANG module "ieee802-dot1q-mstp-bridge.yang" :

- augment “/if:interfaces/if:interface/dot1q:bridge-port” :
  - uses mstp-per-interface-protocol-specific-data

Can be reused in other SDOs like BBF

Base YANG module – MSTP protocol specific YANG definitions as YANG grouping

Per component MSTP YANG data

YANG module "ieee802-dot1q-mstp.yang" :

- YANG “Grouping” (mstp-per-component-protocol-specific-data) :
  - MSTP YANG configuration/state data which are per component specific.

Per interface MSTP YANG data

YANG module "ieee802-dot1q-mstp.yang" :

- YANG “Grouping” (mstp-per-interface-protocol-specific-data) :
  - MSTP YANG configuration/state data which are per interface specific.
IEEE 802.1 Qdy decoupling proposal – MSTP YANG

**Note**: Only the YANG is restructured considering reusability. The resultant YANG tree/functionality is the same as that on page 20 of 802-1Qdy-d2-0.pdf
P802.1Qdy decoupling proposal – MSTP YANG

Resultant YANG tree (IEEE 802.1 Bridge component)

module: ieee802-dot1q-mstp-bridge

augment /dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-mst:
   +-rw mst-config-id!
      |   +-rw format-selector?   int32
      |   +-rw configuration-name? string
      |   +-rw revision-level?    uint32
      |   +-ro configuration-digest?  binary
   +-rw bridge-mstp!
      +-rw max-hops?            int32
      +-ro ist-internal-root-path-cost? uint32
      +-rw mst* [mstid]
         +-rw mstid             uint16
         +-rw port-id-priority? dot1q-types:priority-type
         +-ro internal-root-path-cost? uint32
         +-ro root-port-number?  dot1q-types:port-number-type

The config / state parameters augmented using ieee802-dot1q-mstp.yang grouping.
module: ieee802-dot1q-mstp-bridge

augment /if:interfaces/if:interface/dot1q:bridge-port:
  +--rw port-mstp!
    +--rw mst* [mstid]
        | +--rw mstid
        | +--ro msti-port-state? enumeration
        | +--ro msti-port-role? enumeration
        | +--rw msti-bridge-id-priority? dot1q-types:priority-type
        | +--rw msti-internal-port-path-cost? uint32
        | +--ro msti-regional-root-id? uint32
      +--ro msti-internal-root-path-cost? uint32
      +--ro msti-designated-bridge-id? uint32
      +--ro msti-designated-port-id? uint32

The config / state parameters augmented using ieee802-dot1q-mstp.yang grouping.
IEEE 802.1Qdy decoupling proposal – more details

IEEE 802.1 device specific YANG module which uses Base YANG module

IEEE 802.1Qdy current YANG definition

Can be reused in other SDOs like BBF

Base YANG module – RSTP protocol specific YANG definitions as YANG grouping

YANG module "ieee802-dot1q-rstp-bridge.yang"

- augment "/dot1q:bridges/dot1q:bridge/dot1q:component"
  - uses rstp-per-component-protocol-specific-data

YANG module "ieee802-dot1q-rstp.yang"

- YANG “Grouping” (rstp-per-component-protocol-specific-data)
  - RSTP YANG configuration/state data which are per component specific.

Per component RSTP augment

Per interface RSTP augment

YANG module "ieee802-dot1q-rstp-bridge.yang"

- augment "/if:interfaces/if:interface/dot1q:bridge-port"
  - uses rstp-per-interface-protocol-specific-data

YANG module "ieee802-dot1q-rstp.yang"

- YANG “Grouping” (rstp-per-interface-protocol-specific-data)
  - RSTP YANG configuration/state data which are per interface specific.
IEEE 802.1Qdy decoupling proposal – RSTP YANG

Note: Only the YANG is restructured considering reusability. The resultant YANG tree/functionality is the same as that on page 19 of 802-1Qdy-d2-0.pdf.
BBF Layer-2 Access device: RSTP/MSTP Implementation - possible UML-like view

RSTP/MSTP per component configuration/state data view

BBF specific interface-type on top of IETF interface-type to configure VLAN tag match criteria, other configurations and state data

E.g. ifType ethernetCsMaCd, ieee8023adLag...

Forwarding view: Forwarder possibly modelled to have an additional configuration for the MST Instance ID.
BBF layer-2 access device: RSTP/MSTP interface augmentation (possible view)

YANG module: bbf-dot1q-mstp

```yang
augment "if:interfaces/if:interface" {
  when
    "bbf-dot1q-rstp:rstp and
derived-from-or-self(if:type, 'ianaift:ieee8023adLag') or
derived-from-or-self(if:type, 'ianaift:ethernetCsmacd')"
  { description
    "Augments the link aggregation and ethernet interface
    model with MSTP config and state data.";
  }
  description
  "Augment RSTP Bridge Port with MSTP configuration and
  state data.";
  reference
  "13.24, 13.25, and 13.27 of IEEE Std 802.1Q.";
  uses mstp:mstp-interface-aug-data;
}
```

YANG module: bbf-dot1q-rstp

```yang
augment "if:interfaces/if:interface" {
  when
    "derived-from-or-self(if:type, 'ianaift:ieee8023adLag') or
derived-from-or-self(if:type, 'ianaift:ethernetCsmacd')"
  { description
    "Augments the link aggregation and ethernet interface
    model with RSTP config and state data.";
  }
  description
  "Augment ethernet/lag interface with the RSTP configuration
  and state data.";
  reference
  "13.24, 13.25, and 13.27 of IEEE Std 802.1Q.";
  uses rstp:rstp-interface-aug-data;
}
```

BBF Layer-2 access device could also possibly introduce interface type applicability checks like IEEE Std 802.1Q “interface augment clause with the ‘when’ condition for the ‘bridge-port’ container”.

Page 1783 - IETF interface augment clause with the ‘when’ condition for the ‘bridge-port’ container.
RSTP/MSTP interface augmentation – safe checks

A LAG interface can have more than one ethernet interfaces as member ports.

**Scenario 1**

Step-1 : MSTP configuration applied on LAG1
Member ports : {Ethernet1, Ethernet2}
Step-2 : Trying to apply MSTP configuration on one of its member ports. Example : Ethernet2.

Invalid configuration.

When the YANG for 802.1AX is defined, possibility of introducing safe guards could be considered. Till then or if such a safe guard is not realizable in YANG, then a device implementation would have to take care of such safeguards.

**Scenario 2**

Step-1 : MSTP configuration applied on Ethernet2 which was not part of LAG1.
Step-2 : Trying to associate Ethernet2 interface as member port of LAG1.
P802.1Qdy – RSTP/MSTP resultant YANG files

**YANG modules definition and tree**

File name:
“dy-balraj-reusability-across-sdos-yang-0624-v01.zip”

**Scope:**
- De-coupling proposal for reusability

Includes:
- `ieee802-dot1q-mstp.yang` => MSTP Base module
- `ieee802-dot1q-mstp-bridge.yang` => MSTP IEEE 802.1 specific module
- `ieee802-dot1q-mstp-bridge.tree` => MSTP IEEE 802.1 specific module YANG tree
- `ieee802-dot1q-rstp.yang` => RSTP Base module
- `ieee802-dot1q-rstp-bridge.yang` => RSTP IEEE 802.1 specific module
- `ieee802-dot1q-rstp-bridge.tree` => RSTP IEEE 802.1 specific module YANG tree
- `ieee802-dot1q-bridge.tree` => Resultant IEEE 802.1 Bridge Base YANG tree

No change proposed for the IEEE Base YANG file “`ieee802-dot1q-bridge.yang`”. The ”bridge-mst” container definition considered as such.
Backup slides
These RSTP protocol specific configuration and state data directly augment "/bridges/bridge/component". A non-IEEE 802.1 device cannot import this YANG module due to the direct augmentation.

These RSTP protocol specific interface configuration and state data directly augment "/interface/interface/bridge-port". A non-IEEE 802.1 device cannot import this YANG module due to the direct augmentation.
These MSTP protocol specific configuration and state data directly augment "/bridges/bridge/component".
A non-IEEE 802.1 device cannot import this YANG module due to the direct augmentation.

These MSTP protocol specific interface configuration and state data directly augment "/interface/interface/bridge-port".
A non-IEEE 802.1 device cannot import this YANG module due to the direct augmentation.
IEEE 802.1Qdy decoupling proposal – more details

**ieee802-dot1q-mstp.yang**

YANG “Grouping” - Group
MSTP Protocol
specific configurations and
state data.

A grouping for MSTP
protocol configuration and
state data parameters
which are component
specific.

A grouping for MSTP
protocol configuration and
state data parameters
which are interface specific.

**ieee802-dot1q-rstp.yang**

YANG “Grouping” - Group
RSTP Protocol
specific configurations and
state data.

A grouping for RSTP
protocol configuration and
state data parameters
which are component
specific.

A grouping for RSTP
protocol configuration and
state data parameters
which are interface specific.

**ieee802-dot1q-rstp-bridge.yang**

Augments IEEE 802.1 Bridge
Component/Bridge-port with
RSTP YANG definitions from
ieee802-dot1q-rstp.yang.

**ieee802-dot1q-mstp-bridge.yang**

Augments IEEE 802.1 Bridge
Component/Bridge-port with
MSTP YANG definitions from
ieee802-dot1q-mstp.yang.
module ieee802-dot1q-mstp {

    ..... 
    ..... 

grouping mstp-per-component-protocol-specific-data {

description
"Grouping for MSTP configuration and state data";

    ..... 
    ..... 

} // End of grouping mstp-per-component-protocol-specific-data 

grouping mstp-per-interface-protocol-specific-data {

description
"Grouping for MSTP configuration and state data augment under the interface";

    ..... 
    ..... 

} // End of grouping mstp-per-interface-protocol-specific-data 

}
Augmenting Bridge component/Bridge port with MSTP YANG objects

```yang
module ieee802-dot1q-mstp-bridge {

    augment "dot1q:bridges/dot1q:bridge/dot1q:component/dot1q:bridge-mst" {
        when "../dot1q-rstp:rstp";
        description "Augment RSTP-capable Bridge component with MSTP configuration and management.";
        uses mstp:mstp-per-component-protocol-specific-data;
    }

    augment "if:interfaces/if:interface/dot1q:bridge-port" {
        when "dot1q-rstp:rstp";
        description "Augment RSTP Bridge Port with MSTP configuration";
        reference "13.24, 13.25, and 13.27 of IEEE Std 802.1Q.";
        uses mstp:mstp-per-interface-protocol-specific-data;
    }
}
```
Resultant YANG tree (IEEE 802.1 Bridge component)

module: ieee802-dot1q-rstp-bridge

augment /dot1q:bridges/dot1q:bridge/dot1q:component:
    +--rw rstp!
       +--rw force-protocol-version?   enumeration
       +--ro cist-bridge-id?           uint64
       +--rw cist-bridge-id-priority?  dot1q-types:priority-type
       +--ro cist-root-id?             uint64
       +--ro external-root-path-cost?  uint32
       +--ro cist-root-port-number?    dot1q-types:port-number-type
       +--ro max-age?                  uint8
       +--ro hello-time?                rt-types:timer-value-seconds16
       +--ro forward-delay?            uint8
       +--rw bridge-max-age?           uint8
       +--ro bridge-hello-time?        uint8
       +--rw bridge-forward-delay?     uint8
       +--rw tx-hold-count?            int32
       +--ro migrate-time?             int32
       +--ro time-since-topology-change? uint32
       +--ro topology-change-count?    yang:counter64
P802.1Qdy decoupling proposal – RSTP YANG

Resultant YANG tree (IEEE 802.1 Bridge Port)

module: ieee802-dot1q-rstp-bridge

augment /if:interfaces/if:interface/dot1q:bridge-port:
  +--rw rstp!
      +--ro cist-port-state?   enumeration
      +--ro cist-port-role?    enumeration
      +--ro restricted-role?   boolean
      +--ro restricted-tcn?    boolean
      +--ro cist-port-id?      uint16
      +--rw cist-port-priority?  dot1q-types:priority-type
      +--rw external-port-path-cost?  uint32
      +--ro cist-root-id?      uint32
      +--ro cist-external-path-cost?  uint32
      +--ro designated-bridge-id?  uint32
      +--ro designated-port-id?  binary
      +--rw port-protocol-migration-check?  boolean
      +--rw admin-edge-port?    boolean
      +--ro oper-edge-port?     boolean
      +--rw auto-edge-port?     boolean
      +--rw auto-isolate-port?  boolean
      +--ro isolate-port?      boolean

The config / state parameters augmented from ieee802-dot1q-rstp.yang grouping.
module ieee802-dot1q-rstp {
    ....
    ....
    grouping rstp-per-component-protocol-specific-data {
        description "Grouping for RSTP configuration and state data";
        ....
        ....
    } // End of grouping rstp-per-component-protocol-specific-data

    grouping rstp-per-interface-protocol-specific-data {
        description "Grouping for RSTP configuration and state data augment under the interface";
        ....
        ....
    } // End of grouping rstp-per-interface-protocol-specific-data
}

Base YANG module view
Augmenting Bridge component/Bridge port with RSTP YANG objects

```yamls
module ieee802-dot1q-rstp-bridge {

  augment "/dot1q:bridges/dot1q:bridge/dot1q:component" {
    description   "Augment RSTP configuration and state data.";
    uses rstp:rstp-per-component-protocol-specific-data;
  }

  augment "/if:interfaces/if:interface/dot1q:bridge-port" {
    description    "Augment Bridge Port with RSTP configuration";
    reference     "13.24, 13.25, and 13.27 of IEEE Std 802.1Q.";
    uses rstp:rstp-per-interface-protocol-specific-data;
  }
}
```
### MSTP Topology

#### Access Device 1
(conforming to BBF Layer2 forwarding arch)

- ETH1
- ETH2
- ETH3
- ETH4

#### Access Device 2
(conforming to IEEE Layer2 forwarding arch)

- ETH1
- ETH2
- ETH3
- PON

#### Access Device 3
(conforming to IEEE Layer2 forwarding arch)

- ETH1
- ETH2
- ETH3
- ETH4
- xDSL

#### Interface configuration

- mstid1, mstid2

#### Forwarder 1

**mstp device configuration**

```
mstp-config-id
mstid1 - config
mstid2 - config
```

**Forwarder 1**

**mstid 1**

- Forwarder-ports:
  - VS1 (VLAN 100) \(\rightarrow\) ETH1
  - VS12 (VLAN 100) \(\rightarrow\) ETH2
  - VS13 (VLAN 100) \(\rightarrow\) ETH3

#### Forwarder 2

**mstid 2**

- Forwarder-ports:
  - VS17 (VLAN 102) \(\rightarrow\) ETH1
  - VS18 (VLAN 102) \(\rightarrow\) ETH2
  - VS19 (VLAN 102) \(\rightarrow\) ETH3

#### CIST: Except 100, 101, 102.
All other VLANs part of CIST

#### Access line PHY technology could be xDSL or Ethernet or Fiber

#### ONT

#### PON

#### CPE