

YANG Based VLAN and VID Config for MACsec and MAC Privacy 802.1AEdk

Don Fedyk (dfedyk@labn.net)

Outline

- Forward
- Background Review
- Configuring EDE types with current YANG using Yuma123

Forward

- This presentation is for a discussion on detailed config.
- It may contain errors/omission and should be consider a work in progress.
- An updated version the presentation will be posted after discussion to correct it but it will remain a work in progress.

MACsec

- MACsec is a shim on the leg of a bridge.
- Therefore all VLAN tag configuration just flows through as MACsec does frame by frame authentication and encryption.
- The VLAN tag has the EtherType the priority code point the discard eligible marking and the VLAN ID (VID)
- But in Ethernet Encryption devices the VLAN tagging of frames is implied by component types.
 - Using current IEEE802 YANG there are gaps.
- MAC Privacy also can be collocated with MACsec and has the same behavior with respect to VLAN tag configuration
- MAC Privacy may be physically separate from MACsec and needs both address and VLAN tag controls for this case.

IEEE 802.1Q-2018 Definitions

3.290 Virtual Local Area Network (VLAN): The closure of a set of Media Access Control (MAC) Service Access Points (MSAPs) such that a data request in one MSAP in the set is expected to result in a data indication in another MSAP in the set.

VLAN Config is out of scope for MACsec and MAC privacy

3.294 Virtual Local Area Network (VLAN) tagged frame: A tagged frame whose tag header carries both VLAN identification (VLAN ID or VID) and priority information.

VLAN tags are in scope for EDEs and MAC privacy.

C-VLAN/VID S-VLAN/VID Summary

- The VLAN is the connectivity
- The VLAN Tag is carried on the Frame
- C-TAGs have a TPID – EtherType of 0x8100 & PCP/DE/VID
- S-TAGs have a TPID – EtherType of 0x88A8 & PCP/DE/VID
- One VID is typically used for the control plane RSTP etc.
- Multiple VIDs can map to a single VLAN – this is accomplished with VID and FID mappings – hope not to cover this in MACsec.

Cases handled the Bridge and Provider Bridge using SNMP and YANG

Most Frequent:

- Untagged Frames arrive at a C-Component interface
- C-Tagged Frames arrive/terminate at a C-Component interface
- C-Tagged Frames Map to an S-Component
 - C-tagged frames arrive at a customer network port on an S-vlan-component
- S-Tagged Frames terminate at an S-Component

Less Frequent:

- S-Tagged Frames carry an untagged frame
- S-Tagged Frames arrive at a C-Component



Review: Bridge VLAN Tag Configuration From 802.1Q-2018

EISS shall be supported by using one but not both of the following VLAN tag types:

- a) C-VLAN tag (C-TAG) or
- b) S-VLAN tag (S-TAG)

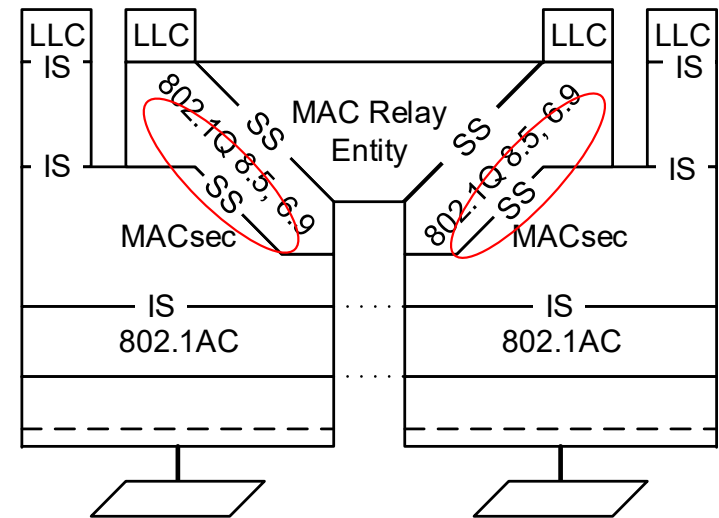
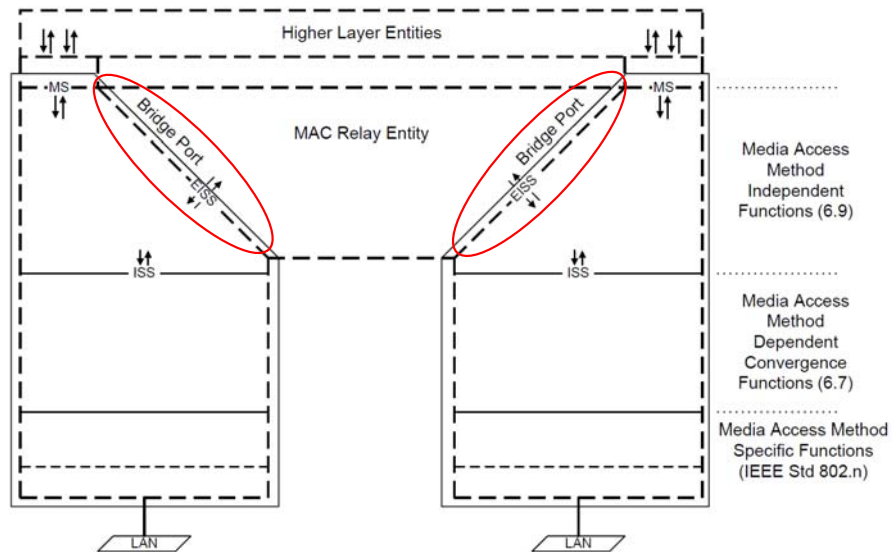
Each Bridge Port shall support the following parameters for use by these functions:

- c) An Acceptable Frame Types parameter with at least one of the following values:
 - 1. Admit Only Untagged and Priority-tagged frames
 - 2. Admit Only VLAN-tagged frames
 - 3. Admit All frames
- d) A PVID parameter for Port-based VLAN classification

Each Bridge Port may support the following parameters:

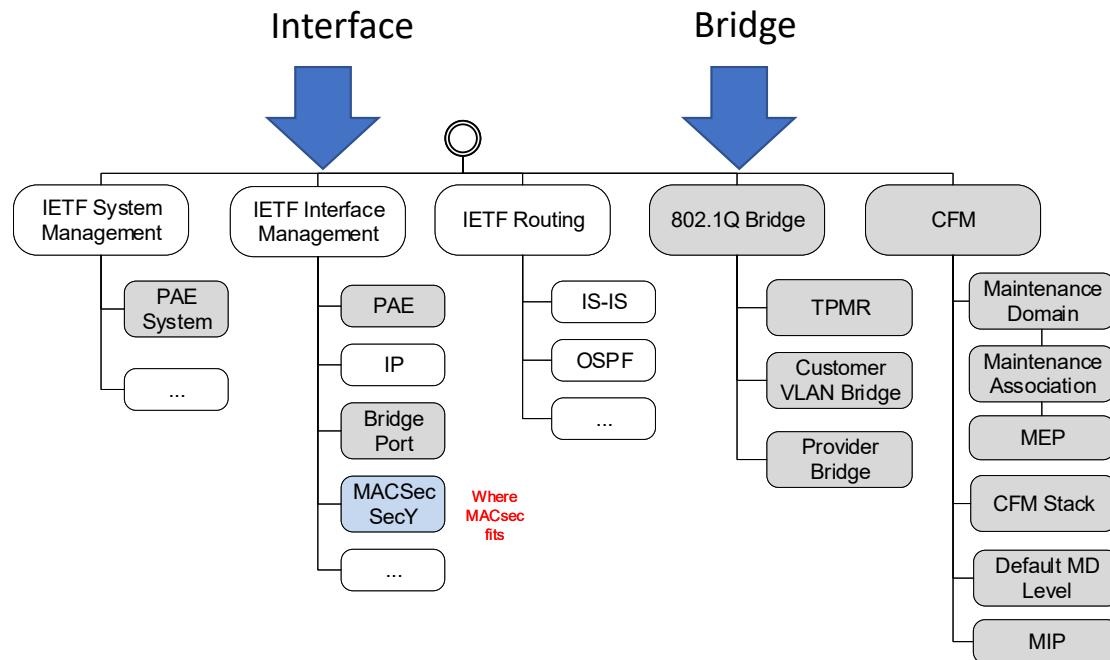
- e) A VID Set for Port-and-Protocol-based classification (6.12)
- f) A VID translation table
- g) An Egress VID translation table

Bridge VLAN tags With and Without MACsec Bridges and Interfaces



MACsec has no specific VLAN Tag Config

In YANG two top level Trees contain VLAN Tag config



EDEs and YANG Summary

- EDE-M
 - VLAN TAG Unaware same YANG
- EDE-CC
 - C-VLAN comes from Bridge and outer C-VID is from inner C-VID
- EDE-CS
 - C-VLAN comes From Bridge and outer S-VID comes from another Bridge but there is a port mapping function.
- EDE-SS



Started here since it follows the standard provider bridge and should have VLAN tag Controls

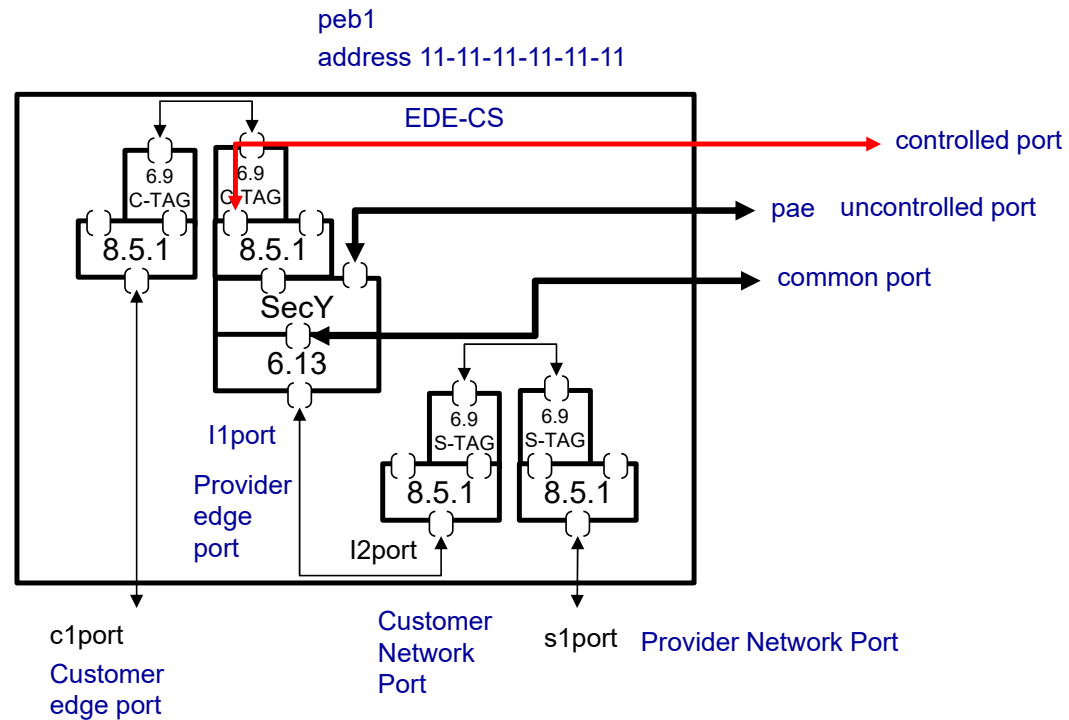
How to do this? Use YUMA123

Yuma123 – Open Source

- Offers a live configuration environment
- Handles multiple YANG modules together
- Validates xpath statements.

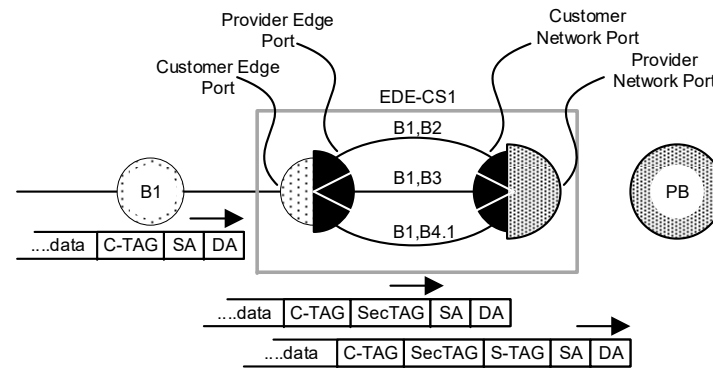
- Parts of the Bridge models due not work in Yuma1232 due to various xpath errors that limited vlan configuration.
- Removing (locally) and eventually correcting xpath allows visibility into what can be configured and to test a complete config.

Revisiting MACsec Config for EDEs



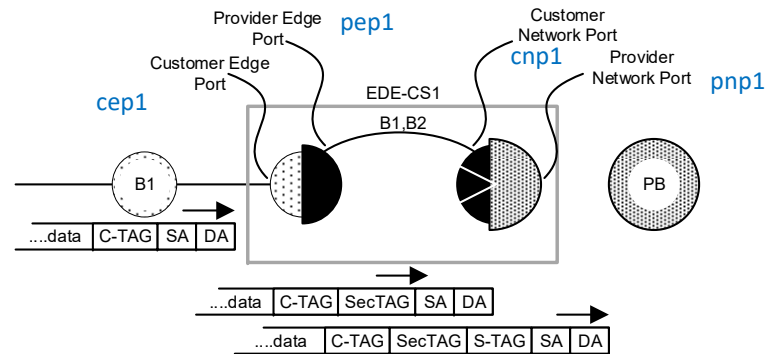
YANG models all these () ports

From 802.1AE-2018 Clause 15.5 CS-EDE



Arriving C-tagged data
Travels Transparent through C-Bridge CEP PEP
S-component adds the S-TAG

First consider a single PEP-CNP mapping



C-TAG – any
S-TAG 200 (or any single legal VID value)

Resulting Config – Is it Complete?

```
rpc-reply {
  data {
    bridges {
      bridge pebridge1 {
        name pebridge1
        address 11-11-11-11-11-11
        bridge-type dot1q:provider-edge-bridge
        component cvlan1 {
          name cvlan1
          type dot1q:c-vlan-component
        }
        component svlan200 {
          name svlan200
          type dot1q:s-vlan-component
          bridge-vlan {
            vlan 200 {
              vid 200
              name svid200
            }
          }
        }
      }
    }
  }
}
interfaces {
  interface cep1 {
    name cep1
    type ianaift:bridge
    bridge-port {
      bridge-name pebridge1
      component-name cvlan1
      port-type dot1q:customer-edge-port
      svid 200
    }
  }
  interface cnp1 {
    name cnp1
    type ianaift:bridge
    bridge-port {
      component-name svlan200
      port-type dot1q:customer-network-port
    }
  }
}
```

c-vlan-component and s-vlan-component

s-vlan-component has a sing VID

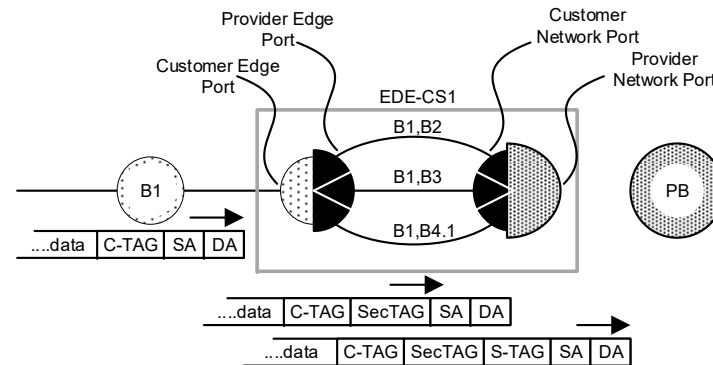
cep is linked to the svid

Note that any additional VLAN config is not illustrated

Questions with the config

- There is an implication when the C-VLAN and the S-VLAN are associated that a certain relationship a binding exists.
- There is all kinds of VLAN related config that is left out.

Now consider Multiple C-VID mapped to SVIDs



Arriving C-TAG -> Maps to SVID
Transparent through C-Bridge CEP PEP
Additional S-TAG

Questions

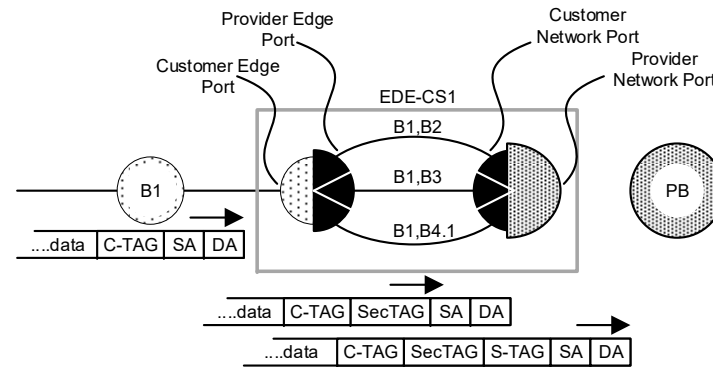
- VID ranges would be beneficial

What about the Reverse Mapping

The Provider Bridging (PB) SNMP has two tables:

- `ieee8021PbCVidRegistrationEntry` that links CVIDs to SVIDs egress the S-component – Just illustrated
- `ieee8021PbEdgePortEntry` an ingress the S-Component when the CVID is Untagged <- This function perhaps is missing in PB YANG?
- Note this situation is unlikely in the case of an CS-EDE this is Provider bridge configuration

Control of egress SVIDs



Provider bridging allows Internal mapping of S-VIDs to External S-VIDs.
Is this used for EDEs? Seems unlikely – so far.

Untagged ports

- The C-VLAN-Component can have Untagged or Tagged ports on the cep for sure.
 - On the pep?
- So can the S-VLAN- Component
 - On the pnp for sure
 - One the cnp ?

A CC- EDE

- A CC-EDE is an Ethernet Data Encryption device
- “The configuration of an EDE-CC is constrained to restrict egress for each Provider Edge Port to a single C-VID and to restrict the PVID for the internally connected Customer Network Port to the same value, with the consequence that the outer C-VID will always match the inner C-VID. The PVID for the Customer EdgePort is constrained to be the same as that for the Provider Network Port and the Static VLAN RegistrationEntry (8.8.2 of IEEE Std 802.1Q-2018) for that and other VIDs are constrained so that frames for that VID are transmitted untagged on both ports, with the consequence that frames received untagged on either ports are forwarded (if at all) untagged on the other.”

Some CC-EDE related config

```
rpc-reply {
  data {
    bridges {
      bridge pebridge1 {
        name pebridge1
        address 11-11-11-11-11-11
        bridge-type dot1q:provider-edge-bridge
        component cvlan1 {
          name cvlan1
          type dot1q:c-vlan-component
          filtering-database {
            filtering-entry 1 200 11-11-11-22-22-22 {
              database-id 1
              vids 200
              address 11-11-11-22-22-22
              entry-type static
              port-map 1 {
                port-ref 1
                static-vlan-registration-entries {
                  registrar-admin-control fixed-new-propagated
                  vlan-transmitted tagged
                }
              }
            }
          }
        }
      }
      bridge-vlan {
        vlan 200 {
          vid 200
          name outer_cc_ede_200
        }
      }
    }
  }
}
```

Is Bridge Type still Provider Edge Bridge? How about a cc-ede-device?

Two c-vlan-components cvlan1 and cvlan2. This is cvlan1

Only one VID Vid 200

Static registration entry

Transmit with TAG

At this point the separation of EDE and Bridging is mixed.

Some CC-EDE related config

```
interface pep1 {  
  name pep1  
  type ianaift:bridge  
  secy {  
  }  
  pae {  
  }  
  bridge-port {  
    bridge-name pebridge1  
    component-name cvlan1  
    port-type dot1q:provider-edge-port  
  }  
}  
interface pnp1 {  
  name pnp1  
  type ianaift:bridge  
  bridge-port {  
    bridge-name pebridge1  
    component-name cvlan2  
    port-type dot1q:provider-network-port  
  }  
}  
}  
nacm {  
}  
system {  
  pae-system {  
    name pae1  
  }  
}  
}
```

This is the provider edge port

It is linked to cvlan2

This is the provider network port

It is linked to cvlan2

What is missing

- There is no configuration that ties the two cvlan components together,
- Two cvlan components can exist with one or more svlan components.
- Suggest an explicit config to tie cvlans together.
- Looks like adding bridge types for cc-edc and cs-edc would be one way.

EDE -SS

- TBD
- Waiting to see if CS-EDE and CC-EDE config is good.

Summary

This presentation has focused on how to configure bridge components to configure various VID tag operations.

This presentation will be updated with the comments from the group.

Backup

- Following is a condensed overview of the Bridge YANG files.
- They may be useful to describe where the relevant config is configured.
- The original YANG that is published is the source.

Bridge Config – VLAN Only Condensed! – Not Complete

```
module: ieee802-dot1q-bridge
+--rw bridges
  +--rw bridge* [name]
    +--rw component* [name]
      +--ro bridge-port* if:interface-ref
      +--ro capabilities
      +--rw filtering-database
        +--rw filtering-entry* [database-id vids address]
          +--rw vids dot1qtypes:vid-range-type
          +--rw port-map* [port-ref]
            +--rw (map-type)?
              +---(static-filtering-entries)
              +---(static-vlan-registration-entries)
              +---(mac-address-registration-entries)
              +---(dynamic-vlan-registration-entries)
              +---(dynamic-reservation-entries)
              +---(dynamic-filtering-entries)
          +--rw vlan-registration-entry* [database-id vids]
            +--rw vids dot1qtypes:vid-range-type
            +--rw port-map* [port-ref]
        +--rw permanent-database
          +--rw filtering-entry* [database-id vids address]
            +--rw vids dot1qtypes:vid-range-type
            +--rw port-map* [port-ref]
      +--rw bridge-vlan
        +--ro max-vids? uint16
        +--ro override-default-pvid? boolean
        +--ro protocol-template? dot1qtypes:protocol-frame-format-type {port-and-protocol-based-vlan}?
        +--ro max-msti? uint16
        +--rw vlan* [vid]
          +--rw vid dot1qtypes:vlan-index-type
          +--rw name? dot1qtypes:name-type
          +--ro untagged-ports* if:interface-ref
          +--ro egress-ports* if:interface-ref
          +--rw protocol-group-database* [db-index] {port-and-protocol-based-vlan}?
          +--rw vid-to-fid-allocation* [vids]
            +--rw vids dot1qtypes:vid-range-type
            +--ro fid? uint32
            +--ro allocation-type? enumeration
          +--rw fid-to-vid-allocation* [fid]
            +--rw fid uint32
            +--ro allocation-type? enumeration
            +--ro vid? dot1qtypes:vlan-index-type
          +--rw vid-to-fid* [vid]
            +--rw vid dot1qtypes:vlan-index-type
            +--ro fid? uint32
      +--rw bridge-mst
        +--rw mstid* dot1qtypes:mstid-type
        +--rw fid-to-mstid* [fid]
          +--rw fid uint32
          +--rw mstid? dot1qtypes:mstid-type
        +--rw fid-to-mstid-allocation* [fids]
          +--rw fids dot1qtypes:vid-range-type
          +--rw mstid? dot1qtypes:mstid-type
```

Filtering database

Control Plane VLAN

STP/RSTP/MSTP/SPB VLAN

Bridge Config – Interface related Condensed! – Not Complete

```
augment /if:interfaces/if:interface:
  +--rw bridge-port
    +--rw bridge-name?          -> /bridges/bridge/name
    +--rw component-name?      -> /bridges/bridge[dot1q:name=current()/../bridge-name]/component/name
    +--rw component-type?      -> /bridges/bridge[dot1q:name=current()/../bridge-name]/component[dot1q:name=current()/../component-name]/type
    +--rw port-type?           identityref
    +--rw pvid?                dot1qtypes:vlan-index-type
    +--rw default-priority?    dot1qtypes:priority-type
    +--rw priority-regeneration
    +--rw enable-ingress-filtering?  boolean
    +--rw enable-restricted-vlan-registration?  boolean
    +--rw enable-vid-translation-table?  boolean
    +--rw enable-egress-vid-translation-table?  boolean
    +--rw protocol-group-vid-set* [group-id] {port-and-protocol-based-vlan}?
      | +--rw group-id  uint32
      | +--rw vid*     dot1qtypes:vlanid
    +--ro protocol-based-vlan-classification?  boolean {port-and-protocol-based-vlan}?
    +--ro max-vid-set-entries?  uint16 {port-and-protocol-based-vlan}?
    +--ro port-number?         dot1qtypes:port-number-type
    +--ro address?             ieee:mac-address
    +--ro capabilities?       bits
    +--ro type-capabilities?  bits
    +--ro external?           boolean
    +--ro oper-point-to-point? boolean
    +--ro statistics
    +--rw vid-translations* [local-vid]
      | +--rw local-vid  dot1qtypes:vlanid
      | +--rw relay-vid? dot1qtypes:vlanid
    +--rw egress-vid-translations* [relay-vid]
      +--rw relay-vid  dot1qtypes:vlanid
      +--rw local-vid? dot1qtypes:vlanid
```

Filtering Related

Port and Protocol Based

VID Translation

Provider Bridge Config – Interface Based

```
module: ieee802-dot1q-pb
augment /if:interfaces/if:interface/dot1q:bridge-port:
  +--rw svid? dot1q-types:vlanid
  +--rw cvid-registration* [cvid]
  |   +--rw cvid dot1q-types:vlanid
  |   +--rw svid? dot1q-types:vlanid
  |   +--rw untagged-pep? boolean
  |   +--rw untagged-cep? boolean
  +--rw service-priority-regeneration* [svid]
  |   +--rw svid dot1q-types:vlanid
  |   +--rw priority-regeneration
  |   |   +--rw priority0? priority-type
  |   |   +--rw priority1? priority-type
  |   |   +--rw priority2? priority-type
  |   |   +--rw priority3? priority-type
  |   |   +--rw priority4? priority-type
  |   |   +--rw priority5? priority-type
  |   |   +--rw priority6? priority-type
  |   |   +--rw priority7? priority-type
  +--rw rcap-internal-interface* [external-svid]
  |   +--rw external-svid dot1q-types:vlanid
  |   +--rw internal-port-number? dot1q-types:port-number-type
  |   +--rw internal-svid? dot1q-types:vlanid
  |   +--rw internal-interface-type? enumeration
```

CVID Registration

Port and Protocol Based

VID Translation