# IEC/IEEE 60802 D2.1 YANG comments #571, #570, and #130

Josef Dorr, Siemens AG

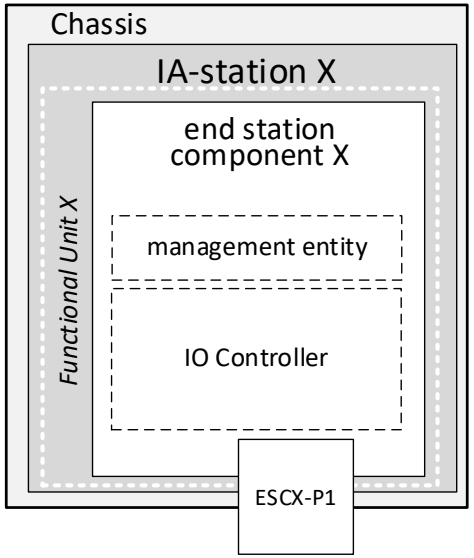
# #571: Is the Bridge-YANG module needed for End Station Components?

See IEC/IEEE 60802 D2.1 comment #571 (https://www.ieee802.org/1/files/private/60802-drafts/d2/60802-d2-1-pdis-v01.pdf):

*“Mandatory values in End Stations: it is unclear why the bridge YANG module is needed and which "Bridge-Type" it should be. Most values have absolutely no meaning such as msti. The overhead for maintaining a filter database does at least at send direction not provide any advantage. There is no clear definition what to do with bridge items in an end station and most end station designs as of today may not have an extended filtering database.*

*Suggested Remedy: Move Bridge Yang module to bridge requirements.”*

# Is the Bridge-YANG module needed for End Station Components?




1. YES, because …

   the Bridge-YANG module provides the nodes to configure: VLANs, Traffic Classes, Transmission Selection, … for bridge and end station components in a uniform way.

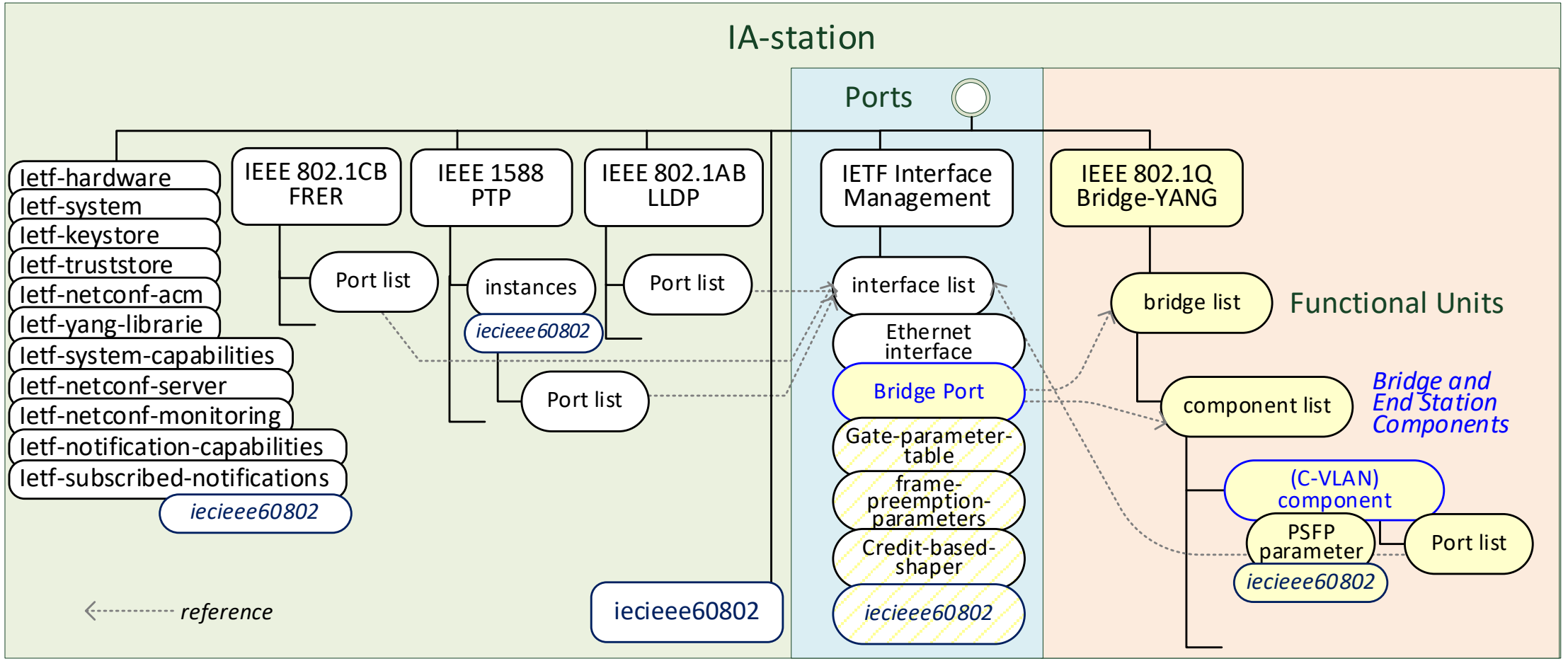
2. Is the node selection for bridge and end station components identical?

   No – the node selection for end station components can be restricted.

3. What Bridge-YANG nodes shall an end station component support?

   See next slides.

# IEC/IEEE 60802 YANG Modules - Overview



# Bridge-YANG node selection for bridge and end station components

(C-VLAN) component

**Bridge Component**

```
+--rw component* [name]
   +--rw name                      string
   +--rw id?                       uint32
   +--rw type                      identityref
   +--rw traffic-class-enabled?    boolean
   +--ro ports?                    uint16
   +--ro bridge-port*              if:interface-ref
   +--ro capabilities              // container
   +--rw filtering-database
   |  +--rw filtering-entry* [database-id vids address]
   |  +--rw vlan-registration-entry* [database-id vids]
   +--rw permanent-database
   |  +--rw filtering-entry* [database-id vids address]
   +--rw bridge-vlan
   +--rw bridge-mst
```

**End Station Component**

```
+--rw component* [name]
   +--rw name                      string
   +--rw id?                       uint32
   +--rw type                      identityref
   +--rw traffic-class-enabled?    boolean
   +--ro ports?                    uint16
   +--ro bridge-port*              if:interface-ref
   +--ro capabilities              // container
   +--rw filtering-database
   |  +--rw filtering-entry* [database-id vids address]
   |  +--rw vlan-registration-entry* [database-id vids]
   +--rw permanent-database
   |  +--rw filtering-entry* [database-id vids address]
   +--rw bridge-vlan               // only vlan list
   +--rw bridge-mst
```

- Only vlan-registration-entries in the filtering database.
- No permanant database
- Only vlan list in bridge-vlan container
- No bridge-mst configuration

# Bridge-YANG node selection for bridge and end station component ports (1)

Bridge Port

| Bridge Component Port | | End Station Component Port | |
|---|---|---|---|
| +--rw bridge-port | | +--rw bridge-port | |
| +--rw bridge-name? | string | +--rw bridge-name? | string |
| +--rw component-name? | string | +--rw component-name? | string |
| +--rw port-type? | | +--rw port-type? | |
| +--rw pvid? | dot1qtypes:vlan-index-type | +--rw pvid? | dot1qtypes:vlan-index-type |
| +--rw default-priority? | dot1qtypes:priority-type | +--rw default-priority? | dot1qtypes:priority-type |
| +--rw traffic-class | // container | +--rw traffic-class | // container |
| +--ro statistics | | +--ro statistics | |

- Identical node selection for the nodes listed in IEC/IEEE 60802 D2.1

**BUT:** Various nodes, which are mandatory for bridge-ports, are missing in the D2.1 Bridge-YANG node selection

transmission-selection-algorithm-table, priority-regeneration, acceptable-frame, enable-ingress-filtering, enable-vid-translation-table, vid-translations, enable-egress-vid-translation-table, egress-vid-translations, …

# Bridge-YANG node selection for bridge and end station component ports (2)

Bridge Port

**Bridge Component Port**

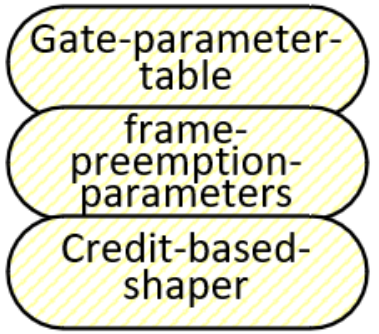
```
+--rw bridge-port
   +--rw bridge-name?       string
   +--rw component-name?    string
   +--rw port-type?
   +--rw pvid?              dot1qtypes:vlan-index-type
   +--rw default-priority?  dot1qtypes:priority-type
   +--rw traffic-class      // container
   +--ro statistics
   +--ro port-number?   dot1qtypes:port-number-type
   +--ro address?           ieee:mac-address
   +--ro capabilities?      bits
   +--ro type-capabilties?  bits
   +--ro external?          boolean
   +--ro media-dependent-overhead?            uint8
   +--rw transmission-selection-algorithm-table
   +--rw priority-regeneration  // table
   +--rw acceptable-frame?                    enumeration
   +--rw enable-ingress-filtering?            boolean
   +--rw enable-vid-translation-table?        boolean
   +--rw vid-translations*        [local-vid]
   +--rw enable-egress-vid-translation-table? boolean
   +--rw egress-vid-translations* [relay-vid]
```

**End Station Component Port**

```
+--rw bridge-port
   +--rw bridge-name?       string
   +--rw component-name?    string
   +--rw port-type?
   +--rw pvid?              dot1qtypes:vlan-index-type
   +--rw default-priority?  dot1qtypes:priority-type
   +--rw traffic-class      // container
   +--ro statistics
   +--ro port-number?   dot1qtypes:port-number-type
   +--ro address?           ieee:mac-address
   +--ro capabilities?      bits
   +--ro type-capabilties?  bits
   +--ro external?          boolean
   +--ro media-dependent-overhead?            uint8
   +--rw transmission-selection-algorithm-table
   +--rw priority-regeneration  // table
   +--rw acceptable-frame?                    enumeration
   +--rw enable-ingress-filtering?            boolean
   +--rw enable-vid-translation-table?        boolean
   +--rw vid-translations*        [local-vid]
   +--rw enable-egress-vid-translation-table? boolean
   +--rw egress-vid-translations* [relay-vid]
```

- No boundary port configuration nodes

# Bridge-YANG node selection for bridge and end station component/port augmentations




- Identical node selection for the features which are mandatory/optional for both bridge component ports and end station component ports:
  - Enhancements for Scheduled Traffic (Gate Parameter Table )
  - Frame Preemption
  - Credit Based Shaper


- PSFP – for bridge components only (see D2.1 comment #129)

  IA-stations which incorporate a bridge component shall support the ieee802-dot1q-psfp-bridge YANG module according to 802.1Qcw …

# #571: Allowed values for digital datasheet Bridge-YANG nodes?

See IEC/IEEE 60802 D2.1 comment #570 (https://www.ieee802.org/1/files/private/60802-drafts/d2/60802-d2-1-pdis-v01.pdf):

*“There are quite a few [m] requirements but it is unclear what allowed values are given in 60802 (e.g. bridge-type, extended-filtering, traffic-classes, static-entry-individual-port, ivl-capable, svl-capable, size, maxmsti).*

*Suggested Remedy: Add a list of allowed values that shall be used.”*

# Allowed values for [m] Bridge-YANG nodes?

## Proposal:

Do not add a list of allowed values, because values

- can either be derived from IEC/IEEE 60802 Clause 5
  (e.g. `/ieee802-dot1q-bridge/bridges/bridge/component/`**`type =`** `c-vlan-component`), or
- should not be restricted
  (e.g. no prohibitions of `/ieee802-dot1q-bridge/bridges/bridge/component/`**`capabilities`**).

# #130: Flow Meter Configuration

See IEC/IEEE 60802 D2.1 comment #130 (https://www.ieee802.org/1/files/private/60802-drafts/d2/60802-d2-1-pdis-v01.pdf):

*“According to 5.7.1 l) it should be possible to assign flow meters to ports;*
*however, the selected nodes [of the ieee802-dot1q-psfp-bridge YANG module] do not allow this assignment.*

*Suggested Remedy: clarify how flow meters can be assigned to ports.”*

# Flow Meter Configuration Model according to IEEE 802.1Q-2022

- Flow meters are traffic filters.
- Traffic specification and port assignment are configured using Stream Filters.

**8.6.5.3 Stream filtering**

A received frame can be associated with a stream filter using the frame's *stream_handle* and *priority* parameters. The *stream_handle* is a sub-parameter of the connection_identifier parameter of the ISS (6.6), provided by the stream identification function specified in IEEE Std 802.1CB.

Each stream filter comprises the following:

a) An integer *stream filter identifier*.
b) A *stream_handle* specification, either:
   1) A single value, as specified in IEEE Std 802.1CB.
   2) A wildcard, that matches any *stream_handle*.
c) A *priority* specification, either:
   1) A single priority value.
   2) A wildcard value that matches any priority value.
d) Maximum SDU size filtering (8.6.5.3.1) information, comprising:
   1) An integer *Maximum SDU size*, in octets. A value of 0 disables maximum SDU size filtering for this stream filter.
   2) A boolean *StreamBlockedDueToOversizeFrameEnable* parameter.
   3) A boolean *StreamBlockedDueToOversizeFrame* parameter.
e) An integer *stream gate identifier* (8.6.5.4).
f) An integer *flow meter instance identifier* (8.6.5.5).
   If this parameter is absent, frames associated with the stream filter are not subject to flow metering.
g) An integer ATS *scheduler instance identifier* (8.6.5.6).
   If this parameter is absent, frames associated with the stream filter are not subject to ATS scheduling and transmission selection.

e.g. with mask-and-match-stream-identification

**8.6.5.5 Flow metering**

The flow meters specified by this subclause (8.6.5.5) implement the parameters and algorithm specified in *Bandwidth Profile Parameters and Algorithm* in MEF 10.3 with the additions described in this subclause.

Each flow meter comprises the following:

a) An integer *flow meter identifier*.
b) An integer *Committed information rate (CIR)*, in bits per second (MEF 10.3).
c) An integer *Committed burst size (CBS)*, in octets (MEF 10.3).
d) An integer *Excess Information Rate (EIR)*, in bits per second (MEF 10.3).
e) An integer *Excess burst size (EBS) per bandwidth profile flow*, in octets (MEF 10.3).
f) A *Coupling flag (CF)*, which takes the value 0 or 1 (MEF 10.3).
g) A *Color mode (CM)*, which takes the value *color-blind* or *color-aware* (MEF 10.3).
h) A boolean *DropOnYellow* parameter.
i) A boolean *MarkAllFramesRedEnable* parameter.
j) A boolean *MarkAllFramesRed* parameter.

# Flow Meters and Stream Filters YANG node selection in IEC/IEEE 60802

NEW

```
+--rw stream-filters
   +--rw stream-filter-instance-table* [stream-filter-instance-id]
   |  +--rw stream-filter-instance-id                    uint32
      +--rw stream-handle                                uint32
      +--rw  flow-meter-ref?
```

```
+--rw flow-meters
|  +--rw flow-meter-instance-table* [flow-meter-instance-id]
|  |  +--rw flow-meter-instance-id         uint32
|  |  +--rw committed-information-rate     uint64
|  |  +--rw committed-burst-size           uint32
|  |  +--rw excess-information-rate        uint64
|  |  +--rw excess-burst-size              uint32
|  |  +--rw coupling-flag                  enumeration
|  |  +--rw color-mode                     enumeration
|  |  +--rw drop-on-yellow                 boolean
|  |  +--rw mark-all-frames-red-enable?    boolean
|  |  +--rw mark-all-frames-red?           boolean
```

NEW

```
+--rw stream-identity* [index]
   +--rw index  uint32
   +--rw handle uint32
   +--rw out-facing
   |  +--rw input-port* if:interface-ref
   +--rw (parameters)
      +--rw mask-and-match-stream-identification
         +--rw destination-mac-mask? ieee802:mac-address
         +--rw destination-mac-match? ieee802:mac-address
```

NOTE For example, an implementation could contain **per out-facing/input-port**:

- one mask and match stream identification for broadcast traffic, and
- one mask and match stream identification for multicast traffic, and
- one mask and match stream identification for unicast traffic.

# Thank You