# Introduction

This proposal is provided by the editor of IEEE P802.1Qcr, Johannes Specht, for discussion during resolution of rogue comment #101 on 802.1Qcr/D1.0. The major motivation of this comment is to enhance clause 48 for extension by new YANG modules of ongoing and future IEEE 802.1 projects, have a common style across the related contents added by such projects to IEEE 802.1Q, and enhance readability due to more symmetry.

Remarks:

a) Editor's Notes are used throughout this proposal for commenting observations, make suggestions, describe alternatives, and similar.
b) Like in IEEE Std 802.1Qcp-2018, there is no strong relationship between the structuring of clause 48.3 of IEEE Std 802.1Qcp-2018 and the structuring found in subsequent clauses of this proposal (see comment #101). However, level 3 subclauses in 48.4 are re-used in this proposal to provide the "glue" in a unified manner. Contents from clause 48.3 are not part of this document, but can be found in IEEE Std 802.1Qcp-2018 and the latest draft of 802.1Qcr.
c) The subdivision of subsequent clauses is close to IEEE 802.1Qcp-2018, although slightly flattened. Contents related YANG modules ieee802-dot1q-tpmr, ieee802-dot1q-vlan-bridge, and ieee802-dot1q-pb are not level 4 subclauses of an enclosing level 3 sub-clause of the ieee802-dot1q-bridge, but are implemented as subsequent level 3 subclauses instead.
While the level 3 subclauses in the subsequent shown level 2 subclauses 48.5, 48.6, and 48.7 are symmetrically structured (one level 3 subclause per YANG module), an alternative would be to bundle the level 3 subclauses of a particular module together.
d) Level 3 subclauses for which the content was not available during creation of this document are marked with "N/A". This content may be provided via maintenance, contributions, or similar.

# 48. YANG Data Model

## 48.4 Structure of the YANG model

IEEE 802.1Q YANG models are divided into a number of YANG modules. A summary of the modules contained in this clause is represented in Table 48-1.

**Table 48-1—Summary of YANG modules**

| Module | References | Description |
|---|---|---|
| ieee802-types | 48.5.1, 48.6.1, 48.7.1 | General type definitions used within IEEE 802 standards. |
| ieee802-dot1q-types | 48.5.2, 48.6.2, 48.7.2 | General type definitions used by IEEE 802.1Q standard. |
| ieee802-dot1q-bridge | 48.5.3, 48.6.3, 48.7.3 | Generic IEEE 802.1Q Bridge YANG model, which is augmented by specific IEEE 802.1Q bridges. |
| ieee802-dot1q-tpmr | 48.5.4, 48.6.4, 48.7.4 | Two-Port MAC Relay YANG model, which augments the generic bridge YANG module. |
| ieee802-dot1q-vlan-bridge | 48.5.5, 48.6.5, 48.7.5 | Customer VLAN Bridge YANG model, which augments the generic bridge YANG module. |
| ieee802-dot1q-pb | 48.5.6, 48.6.6, 48.7.6 | Provider Bridges YANG model, which augments the generic bridge YANG module. |
| ieee802-dot1q-stream-filters-gates | 48.5.7, 48.6.7, 48.7.7 | Stream Filters and Stream Gates common to all applications in 8.6.5.4. |
| ieee802-dot1q-ats | 48.5.8, 48.6.8, 48.7.8 | ATS-specific extensions to the ieee802-dot1q-stream-filters-gates and ieee802-dot1q-bridge modules. |

The relationship between the models listed in clause 48.3 and the YANG modules listed in Table 48-1 is described in the following subclauses.

<<Editor's Note: The new 48.4.x subclauses may be referred to from conformance clauses, etc. The "Notes" column in subsequently shown tables may be used to describe partial implementation (e.g., full implementation of ieee802-types is typically not required). If such details are not needed, the tables could be replaced by lettered lists.>>

### 48.4.1 Generic Bridge model

The generic Bridge model provides basic bridging capabilities and allows for augmentation by specific YANG models (e.g., Two-Port MAC Relay model, Customer Bridge YANG model, Provider Bridge model).

A system implementing the generic Bridge model implements the YANG modules as described Table 48-3.

**Table 48-2—YANG module dependencies for the generic Bridge model**

| YANG module | Notes |
|---|---|
| *ieee802-types* | N/A |
| *ieee802-dot1q-types* | N/A |
| *ieee802-dot1q-bridge* | N/A |

### 48.4.2 Two-Port MAC Relay model

A system implementing the TPMR YANG model (48.3.2.1) implements the YANG modules as described Table 48-3.

**Table 48-3—YANG module dependencies for the Two-Port MAC Relay model**

| YANG module | Notes |
|---|---|
| *ieee802-types* | N/A |
| *ieee802-dot1q-types* | N/A |
| *ieee802-dot1q-bridge* | N/A |
| *ieee802-dot1q-tpmr* | N/A |

### 48.4.3 Customer VLAN Bridge model

A system implementing the Customer VLAN Bridge YANG model (48.3.2.2) implements the YANG modules as described in Table 48-4.

**Table 48-4—YANG module dependencies for the Customer VLAN Bridge model**

| YANG module | Notes |
|---|---|
| *ieee802-types* | N/A |
| *ieee802-dot1q-types* | N/A |
| *ieee802-dot1q-bridge* | N/A |
| *ieee802-dot1q-vlan-bridge* | N/A |

### 48.4.4 Provider Bridge model

A system implementing the Provider Bridge YANG model (48.3.2.3) implements the YANG modules as described in Table 48-5.

**Table 48-5—YANG module dependencies for the Provider Bridge model**

| YANG module | Notes |
|---|---|
| *ieee802-types* | N/A |
| *ieee802-dot1q-types* | N/A |
| *ieee802-dot1q-bridge* | N/A |
| *ieee802-dot1q-pb* | N/A |

### 48.4.5 Stream Filter and Stream Gates model

The Stream Filter and Stream Gates model (48.3.3) provides basic stream filter (8.6.5.1) and stream gate (8.6.5.2) capabilities and allows for augmentation by specific YANG models (e.g., ATS model).

A system implementing the Stream Filter and Stream Gates model implements the YANG modules as described Table 48-6.

**Table 48-6—YANG module dependencies for the Stream Filter and Stream Gates model**

| YANG module | Notes |
|---|---|
| *ieee802-types* | N/A |
| *ieee802-dot1q-types* | N/A |
| *ieee802-dot1q-bridge* | N/A |
| *ieee802-dot1q-stream-filters-gates* | N/A |

### 48.4.6 Asynchronous Traffic Shaping (ATS) model

A system implementing the ATS model (48.3.4) implements the YANG modules as described Table 48-7.

**Table 48-7—YANG module dependencies for the Stream Filter and Stream Gates model**

| YANG module | Notes |
|---|---|
| *ieee802-types* | N/A |
| *ieee802-dot1q-types* | N/A |
| *ieee802-dot1q-bridge* | N/A |
| *ieee802-dot1q-stream-filters-gates* | N/A |
| *ieee802-dot1q-ats* | N/A |

## 48.5 Relationship to IEEE 802.1Q managed objects

<<Editor's Note: The subsequent introduction copied from IEEE Std 802.1Qcp-2018 and needs adjustments>>

This standard specifies a Unified Modeling Language (UML) [B78] information model and a YANG data model that allows configuration and status reporting for bridges and bridge components including Media Access Control (MAC) Bridges, Two-Port MAC Relays (TPMRs), Customer Virtual Local Area Network (VLAN) Bridges, and Provider Bridges (as specified by this standard) with the capabilities currently specified in 12.4 to 12.8, 12.10, 12.13, and 12.19 of this standard.

In support of this standard, the YANG data model extends the IETF Interface Management YANG model (as specified in IETF RFC 8343).

The Bridge Port YANG node augments the Interface Management YANG model. The specific Bridge (e.g., TPMR, Customer VLAN, Provider Bridge) YANG models are augmentations from the Bridge YANG model. A system implementing these YANG models shall implement the *ieee802-dot1q-bridge*, *ieee802-types*, and *ieee802-dot1q-types* YANG models.

### 48.5.1 Relationship of the ieee802-types YANG module

N/A

<<Editor's Note: The treatment of ieee802-types YANG module needs discussion (e.g., scope?!), given that it contains IEEE802-wide types. However, These are simple and we may omit descriptions for these in this clause.>>

### 48.5.2 Relationship of the ieee802-dot1q-types YANG module

N/A

<<Editor's Note: Contents of ieee802-dot1q-types are simple, we may omit descriptions for these.in this clause. Alternatively (which may even be better), we may craft an explicit text that can be copied into all new level 3 subclauses of this clause (48.5) in case the relationship table is not provided (i.e., stating the criteria, etc.)>>

### 48.5.3 Relationship of the ieee802-dot1q-bridge YANG module

<<I've made some minor tweaks, however, the base content is copied from multiple tables in IEEE 802.1Qcp-2018. The tweaks are as follows:
- read/write attributes ("r-w" and "r") removed (already visible at other places and thus less prone to break, MIB tables don't show these either)
- slight adjustments to the bold cell contents in the left column
- Indentations simplified>>

**Table 48-8—Cross-reference table of the ieee802-dot1q-bridge YANG module**

| Bridge management information | YANG node(s) |
|---|---|
| *** Bridge** | **ieee802-dot1q-bridge:bridges:bridge** |
| name (12.4) | name — KEY |
| address (12.4) | address |
| type | bridge-type |
| ports (12.4) | ports |
| upTime (12.4) | up-time |
| components (12.3) | components |
| *** Bridge Component** | **ieee802-dot1q-bridge:bridges:bridge:component** |
| — | name — KEY |
| id (12.3) | id |
| type (12.3) | type |
| address (8.13.8, 13.24) | address |
| trafficClassEnabled (12.4.1.5.1) | traffic-class-enabled |
| ports (12.4.1.1.3) | ports |
| * bridgePorts (—) | * bridge-ports |
| **Bridge Component Capabilities** | **ieee802-dot1q-bridge:bridges:bridge:component:capabilities** |
| extendedFiltering (12.4.1.5.2) | extended-filtering |
| trafficClasses (12.4.1.5.2) | traffic-classes |
| staticEntryIndividualPort (12.4.1.5.2) | static-entry-individual-port |
| ivlCapable (12.4.1.5.2) | ivl-capable |
| svlCapable (12.4.1.5.2) | svl-capable |
| hybridCapable (12.4.1.5.2) | hybrid-capable |
| configurablePvidTagging (12.4.1.5.2) | configurable-pvid-tagging |
| localVlanCapable (12.4.1.5.2) | local-vlan-capable |

**Table 48-8—Cross-reference table of the ieee802-dot1q-bridge YANG module *(continued)***

| Bridge management information | YANG node(s) |
|---|---|
| **Filtering Database** | **ieee802-dot1q-bridge:bridges:bridge:component:filtering-database** |
| agingTime (12.7, 8.8.3) | aging-time |
| size (12.7) | size |
| staticEntries (12.7, 8.8.1) | static-entries |
| dynamicEntries (12.7, 8.8.3) | dynamic-entries |
| staticVlanRegistrationEntries (12.7, 8.8.2) | static-vlan-registration-entries |
| dynamicVlanRegistrationEntries (12.7, 8.8.5) | dynamic-vlan-registration-entries |
| macAddressRegistrationEntries (12.7, 8.8.4) | mac-address-registration-entries |
| **Filtering Entries** | **ieee802-dot1q-bridge:bridges:bridge:component:filtering-database:filtering-entries** |
| databaseId (12.7.7) | database-id — KEY |
| address (12.7.7) | address — KEY |
| vid (12.7.7) | vid — KEY |
| entryType (12.7.7) | entry-type |
| portMap (8.8.1, 8.8.2) | port-map |
| status (—) | status |
| **VLAN Registration Entries** | **ieee802-dot1q-bridge:bridges:bridge:component:filtering-database:vlan-registration-entries** |
| databaseId (12.7.7) | database-id — KEY |
| vid (12.7.7) | vid — KEY |
| entryType (12.7.7) | entry-type |
| portMap (8.8.1, 8.8.2) | port-map |
| **Permanent Database** | **ieee802-dot1q-bridge:bridges:bridge:component:permanent-database** |
| size (12.7.6) | size |
| staticEntries (12.7.6) | static-entries |
| staticVlanRegistrationEntries (12.7.6) | static-vlan-registration-entries |
| **Permanent Filtering Entries** | **ieee802-dot1q-bridge:bridges:bridge:component:permanent-database:filtering-entries** |
| databaseId (12.7.7) | database-id — KEY |
| address (12.7.7) | address — KEY |
| vid (12.7.7) | vid — KEY |
| portMap (8.8.1, 8.8.2) | port-map |

**Table 48-8—Cross-reference table of the ieee802-dot1q-bridge YANG module *(continued)***

| Bridge management information | YANG node(s) |
|---|---|
| **Bridge VLAN** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-vlan** |
| version (12.10.1.3) | version |
| maxVids (12.10.1.3) | max-vids |
| overrideDefaultPvid (12.10.1.3) | override-default-pvid |
| protocolTemplate (12.10.1.7) | protocol-template |
| maxMsti (12.10.1.7) | max-msti |
| **Bridge VLAN ID Entries** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-vlan:vlan-id** |
| vid (12.10.2) | vid — KEY |
| name (12.10.2) | name |
| vid (12.10.2) | vid |
| * untaggedPorts (8.8.2, 12.10.2.1.3) | * untagged-ports |
| * egressPorts (8.8.10, 12.10.2.1.3) | * egress-ports |
| **Protocol Group Database** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-vlan:protocol-group-database** |
| frameFormatType (12.10.1.7) | frame-format-type |
| protocolGroupId (6.12.2) | protocol-group-id |
| **VID to FID** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-vlan:vid-to-fid** |
| vid (12.10.3.4) | vid |
| fid (12.10.3.4) | fid |
| **VID to FID Allocations** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-vlan:vid-to-fid-allocation** |
| vid (12.10.3.2) | vid — KEY |
| fid (12.10.3).2 | fid |
| allocationType (12.10.3.2) | allocation-type |
| **FID to VID Allocations** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-vlan:fid-to-vid-allocation** |
| fid (12.10.3.3) | fid — KEY |
| * vid (12.10.3) | * vid |
| * allocationType (12.10.3) | * allocation-type |

**Table 48-8—Cross-reference table of the ieee802-dot1q-bridge YANG module *(continued)***

| Bridge management information | YANG node(s) |
|---|---|
| **Bridge MST** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-mst** |
| *** MSTID (12.12.1)** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-mst:mstid** |
| **FID to MSTID** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-mst:fid-to-mstid** |
| fid (12.12.2) | fid — KEY |
| mstid (12.12.2) | mstid |
| **FID to MSTID Allocation** | **ieee802-dot1q-bridge:bridges:bridge:component:bridge-mst:fid-to-mstid-allocation** |
| fids (12.12.2) | fids — KEY |
| mstid (12.12.2) | mstid |

**Table 48-8—Cross-reference table of the ieee802-dot1q-bridge YANG module *(continued)***

| Bridge management information | YANG node(s) |
|---|---|
| **Bridge Port** | **ietf-interfaces:interfaces:interface:ieee802-dot1q-bridge:bridge-port** |
| componentName | component-name |
| pvid (5.4, 12.10.1) | pvid |
| defaultPriority (12.6.2) | default-priority |
| priorityRegenerationTable (12.6.2, 6.9.4) | priority-regeneration-table |
| pcpSelection (12.6.2, 6.9.3) | pcp-selection |
| pcpDecodingTable (12.6.2, 6.9.3) | pcp-decoding-table |
| pcpEncodingTable (12.6.2) | pcp-encoding-table |
| useDei (12.6.2, 6.9.3) | use-dei |
| dropEncoding (12.6.2, 8.6.6) | drop-encoding |
| serviceAccessPrioritySelection (12.6.2, 6.13) | service-access-priority-selection |
| serviceAccessPriority (12.6.2, 6.13.1) | service-access-priority |
| trafficClass (112.6.3, 8.6.6) | traffic-class |
| acceptableFrame (12.10.1.3, 6.9) | acceptable-frame |
| enableIngressFiltering (12.10.1.4, 8.6.2) | enable-ingress-filtering |
| restrictedVlanRegistration (12.10.1.6, 11.2.3.2.3) | enable-restricted-vlan-registration |
| vidTranslationTable (12.10.1.8, 6.9) | enable-vid-translation-table |
| egressVidTranslationTable (12.10.1.9, 6.9) | enable-egress-vid-translation-table |
| protocolGroupId (6.12.2) | protocol-group-id |
| protocolGroupDatabaseContents (12.10.1.7) | protocol-group-vid-set |
| adminPointToPoint (6.8.2, 12.4.2) | admin-point-to-point |
| * vidTranslations (12.10.1.8, 6.9) | * vid-translations |
| * egressVidTranslations (12.10.1.9, 6.9) | * egress-vid-translations |
| protocolBasedVlanClassification (5.4.1.2) | protocol-based-vlan-classification |
| maxVidSetEntries (12.10.1.1.3) | max-vid-set-entries |
| portNumber (13.25, 12.4.2) | port-number |
| portType (12.4.2.1) | port-type |
| address (12.4.2) | address |
| capabilities (12.4.2, 12.10.1.1.3) | capabilities |
| typeCapabilities (12.4.2) | type-capabilities |
| external (12.4.2) | external |
| operPointToPoint (12.4.2) | oper-point-to-point |
| mediaDependentOverhead (12.4.2) | media-dependent-overhead |

**Table 48-8—Cross-reference table of the ieee802-dot1q-bridge YANG module *(continued)***

| Bridge management information | YANG node(s) |
|---|---|
| **Bridge Port Statistics** | **ietf-interfaces:interfaces:interface:ieee802-dot1q-bridge:bridge-port:statistics** |
| delayExceededDiscard (12.6.1.1.3, 8.6.6) | delay-exceeded-discard |
| mtuExceededDiscards (12.6.1.1.3) | mtu-exceeded-discards |
| frameRx (12.6.1.1.3) | frame-rx |
| octetsRx (12.6.1.1.3) | octets-rx |
| frameTx () | frame-tx |
| octetsTx () | octets-tx |
| discardInbound (12.6.1.1.3) | discard-inbound |
| forwardOutbound (12.6.1.1.3) | forward-outbound |
| discardLackOfBuffers (12.6.1.1.3) | discard-lack-of-buffers |
| discardTransitDelayExceeded (12.6.1.1.3) | discard-transit-delay-exceeded |
| discardOnError (12.6.1.1.3) | discard-on-error |
| discardOnIngressFiltering (12.6.1.1.3) | discard-on-ingress-filtering |

### 48.5.4 Relationship of the ieee802-dot1q-tpmr YANG module

N/A

### 48.5.5 Relationship of the ieee802-vlan-bridge YANG module

N/A

### 48.5.6 Relationship of the ieee802-dot1q-pb YANG module

N/A

### 48.5.7 Relationship of the ieee802-dot1q-stream-filters-gates YANG module

**Table 48-13—Cross-reference table of the ieee802-dot1q-stream-filters-gates YANG module**

| Bridge management information | YANG node(s) |
|---|---|
| **Stream Filters** | **ieee802-dot1q-bridge:bridges:bridge:component:stream-filters** |
| MaxStreamFilterInstances (12.31.1.1) | max-stream-filter-instances |
| **Stream Filter Instance Table (Table 12-30)** | **ieee802-dot1q-bridge:bridges:bridge:component:stream-filters:stream-filter-instance-table** |
| StreamFilterInstance (12.31.2.1) | stream-filter-instance-id—KEY |
| StreamHandleSpec (12.31.2.2) | stream-handle-spec |
| PrioritySpec (12.31.2.3) | priority-spec |
| StreamGateInstanceID (Table 12-30) | stream-gate-ref |
| **FilterSpecificationList (12.31.2.5)** | **ieee802-dot1q-bridge:bridges:bridge:component:stream-filters:stream-filter-instance-table:filter-specification-list** |
| — | index—KEY |
| **Maximum SDU Size Filters** | **ieee802-dot1q-bridge:bridges:bridge:component:stream-filters:stream-filter-instance-table:filter-specification-list:maximum-sdu-size** |
| An Integer value representing a Maximum SDU size (12.31.2.5, item a) | maximum-sdu-size |
| StreamBlockedDueToOversizeFrameEnable (12.31.2) | stream-blocked-due-to-oversize-frame-enabled |
| StreamBlockedDueToOversizeFrame (12.31.2) | stream-blocked-due-to-oversize-frame |
| **Stream Gates** | **ieee802-dot1q-bridge:bridges:bridge:component:stream-gates** |
| MaxStreamGateInstances (12.31.1.2) | max-stream-gate-instances |
| **Stream Gate Instance Table (Table 12-31)** | **ieee802-dot1q-bridge:bridges:bridge:component:stream-gates:stream-gate-instance-table** |
| StreamGateInstance (12.31.2.4) | stream-gate-instance-id—KEY |
| StreamGateEnabled (Table 12-31) | stream-handle-spec |
| PrioritySpec (Table 12-31,12.31.2.3) | priority-spec |
| AdminGateStates (Table 12-31) | admin-gate-states |
| AdminIPV (12.31.2.5) | admin-ipv |

### 48.5.8 Relationship of the ieee802-dot1q-ats YANG module

**Table 48-14—Cross-reference table of the ieee802-dot1q-ats YANG module**

| Generic bridge management information | YANG node(s) |
|---|---|
| **ATS Schedulers** | **ieee802-dot1q-bridge:bridges:bridge:component:schedulers** |
| MaxSchedulerInstances (12.31.1.5) | max-scheduler-instances |
| **Scheduler Instance Table (Table 12-33)** | **ieee802-dot1q-bridge:bridges:bridge:component:schedulers:scheduler-instance-table** |
| SchedulerInstanceID (12.31.5.1) | scheduler-instance-id—KEY |
| CommittedBurstSize (12.31.5.2) | committed-burst-size |
| CommittedInformationRate (12.31.5.3) | committed-information-rate |
| SchedulerGroupInstanceID (12.31.5.4) | scheduler-group-ref |
| **ATS Scheduler Groups** | **ieee802-dot1q-bridge:bridges:bridge:component:scheduler-groups** |
| MaxSchedulerGroupInstances (12.31.1.6) | max-scheduler-group-instances |
| **Scheduler Group Instance Table (Table 12-34)** | **ieee802-dot1q-bridge:bridges:bridge:component:scheduler-groups:scheduler-group-instance-table** |
| SchedulerGroupInstanceID (12.31.6.1) | scheduler-group-instance-id—KEY |
| MaxResidenceTime (12.31.6.2) | max-residence-time |
| **ATS Scheduler Timing Characteristics Table (Table 12-36)** | **ieee802-dot1q-bridge:bridges:bridge:component:scheduler-timing-characteristics:scheduler-timing-characteristics-table** |
| ReceptionPortNumber (12.31.8.1) | reception-port—KEY |
| TransmissionPortNumber (12.31.8.2) | transmission-port—KEY |
| ClockOffsetVariationMax (12.31.8.3) | clock-offset-variation-max |
| ClockRateDeviationMax (12.31.8.4) | clock-rate-deviation-max |
| ArrivalRecognitionDelayMax (12.31.8.5) | arrival-recognition-delay-max |
| ProcessingDelayMin (12.31.8.6) | processing-delay-min |
| ProcessingDelayMax (12.31.8.7) | processing-delay-max |

## 48.6 YANG data scheme tree definitions

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration (read-write), and "ro" means state data (read-only).
- Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

Ellipsis ("...") stands for contents of subtrees that are not shown.

### 48.6.1 Data scheme definition for the ieee802-types YANG module

N/A

### 48.6.2 Data scheme definition for the ieee802-dot1q-types YANG module

N/A

### 48.6.3 Data scheme definition for the ieee802-dot1q-bridge YANG module

```
module: ieee802-dot1q-bridge
    +--rw bridges
       +--rw bridge* [name]
          +--rw name           dot1qtypes:name-type
          +--rw address        ieee:mac-address
          +--rw bridge-type    identityref
          +--ro ports?         uint16
          +--ro up-time?       yang:zero-based-counter32
          +--ro components?    uint32
          +--rw component* [name]
             +--rw name                      string
             +--rw id?                       uint32
             +--rw type                      identityref
             +--rw address?                  ieee:mac-address
             +--rw traffic-class-enabled?    boolean
             +--ro ports?                    uint16
             +--ro bridge-port*              if:interface-ref
             +--ro capabilities
             |  +--ro extended-filtering?             boolean
             |  +--ro traffic-classes?                boolean
             |  +--ro static-entry-individual-port?   boolean
             |  +--ro ivl-capable?                    boolean
             |  +--ro svl-capable?                    boolean
             |  +--ro hybrid-capable?                 boolean
             |  +--ro configurable-pvid-tagging?      boolean
             |  +--ro local-vlan-capable?             boolean
             +--rw filtering-database
             |  +--rw aging-time?                           uint32
             |  +--ro size?                                 yang:gauge32
             |  +--ro static-entries?                       yang:gauge32
             |  +--ro dynamic-entries?                      yang:gauge32
             |  +--ro static-vlan-registration-entries?     yang:gauge32
             |  +--ro dynamic-vlan-registration-entries?    yang:gauge32
          |  +--ro mac-address-registration-entries?    yang:gauge32 {extended-
filtering-services}?
             |  +--rw filtering-entry* [database-id vids address]
             |  |  +--rw database-id    uint32
             |  |  +--rw address        ieee:mac-address
             |  |  +--rw vids           dot1qtypes:vid-range-type
```

```
|   |  +--rw entry-type?    enumeration
|   |  +--rw port-map* [port-ref]
|   |  |  +--rw port-ref                              port-number-type
|   |  |  +--rw (map-type)?
|   |  |     +--:(static-filtering-entries)
|   |  |     |  +--rw static-filtering-entries
|   |  |     |     +--rw control-element?         enumeration
|   |  |     |     +--rw connection-identifier?   port-number-type
|   |  |     +--:(static-vlan-registration-entries)
|   |  |     |  +--rw static-vlan-registration-entries
|   |  |     |     +--rw registrar-admin-control?   enumeration
|   |  |     |     +--rw vlan-transmitted?          enumeration
|   |  |     +--:(mac-address-registration-entries)
|   |  |     |  +--rw mac-address-registration-entries
|   |  |     |     +--rw control-element?   enumeration
|   |  |     +--:(dynamic-vlan-registration-entries)
|   |  |     |  +--rw dynamic-vlan-registration-entries
|   |  |     |     +--rw control-element?   enumeration
|   |  |     +--:(dynamic-reservation-entries)
|   |  |     |  +--rw dynamic-reservation-entries
|   |  |     |     +--rw control-element?   enumeration
|   |  |     +--:(dynamic-filtering-entries)
|   |  |        +--rw dynamic-filtering-entries
|   |  |           +--rw control-element?   enumeration
|   |  +--ro status?        enumeration
|   +--rw vlan-registration-entry* [database-id vids]
|      +--rw database-id    uint32
|      +--rw vids           dot1qtypes:vid-range-type
|      +--rw entry-type?    enumeration
|      +--rw port-map* [port-ref]
|         +--rw port-ref                              port-number-type
|         +--rw (map-type)?
|            +--:(static-filtering-entries)
|            |  +--rw static-filtering-entries
|            |     +--rw control-element?         enumeration
|            |     +--rw connection-identifier?   port-number-type
|            +--:(static-vlan-registration-entries)
|            |  +--rw static-vlan-registration-entries
|            |     +--rw registrar-admin-control?   enumeration
|            |     +--rw vlan-transmitted?          enumeration
|            +--:(mac-address-registration-entries)
|            |  +--rw mac-address-registration-entries
|            |     +--rw control-element?   enumeration
|            +--:(dynamic-vlan-registration-entries)
|            |  +--rw dynamic-vlan-registration-entries
|            |     +--rw control-element?   enumeration
|            +--:(dynamic-reservation-entries)
|            |  +--rw dynamic-reservation-entries
|            |     +--rw control-element?   enumeration
|            +--:(dynamic-filtering-entries)
|               +--rw dynamic-filtering-entries
|                  +--rw control-element?   enumeration
+--rw permanent-database
|  +--ro size?                                yang:gauge32
|  +--ro static-entries?                      yang:gauge32
|  +--ro static-vlan-registration-entries?   yang:gauge32
|  +--rw filtering-entry* [database-id vids address]
|     +--rw database-id    uint32
|     +--rw address        ieee:mac-address
|     +--rw vids           dot1qtypes:vid-range-type
|     +--ro status?        enumeration
|     +--rw port-map* [port-ref]
|        +--rw port-ref                              port-number-type
|        +--rw (map-type)?
|           +--:(static-filtering-entries)
|           |  +--rw static-filtering-entries
|           |     +--rw control-element?         enumeration
|           |     +--rw connection-identifier?   port-number-type
|           +--:(static-vlan-registration-entries)
|           |  +--rw static-vlan-registration-entries
|           |     +--rw registrar-admin-control?   enumeration
|           |     +--rw vlan-transmitted?          enumeration
```

```
               |           +--:(mac-address-registration-entries)
               |           |  +--rw mac-address-registration-entries
               |           |     +--rw control-element?   enumeration
               |           +--:(dynamic-vlan-registration-entries)
               |           |  +--rw dynamic-vlan-registration-entries
               |           |     +--rw control-element?   enumeration
               |           +--:(dynamic-reservation-entries)
               |           |  +--rw dynamic-reservation-entries
               |           |     +--rw control-element?   enumeration
               |           +--:(dynamic-filtering-entries)
               |              +--rw dynamic-filtering-entries
               |                 +--rw control-element?   enumeration
               +--rw bridge-vlan
               |  +--ro version?                   uint16
               |  +--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 db-index             uint16
               |  |  +--rw frame-format-type?   dot1qtypes:protocol-frame-format-
type
               |  |  +--rw (frame-format)?
               |  |  |  +--:(ethernet-rfc1042-snap8021H)
               |  |  |  |  +--rw ethertype?           dot1qtypes:ethertype-type
               |  |  |  +--:(snap-other)
               |  |  |  |  +--rw protocol-id?         string
               |  |  |  +--:(llc-other)
               |  |  |     +--rw dsap-ssap-pairs
               |  |  |        +--rw llc-address?   string
               |  |  +--rw group-id?            uint32
               |  +--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
               |     +--rw 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
  augment /if:interfaces/if:interface:
    +--rw bridge-port
       +--rw component-name?                        string
       +--rw port-type?                             identityref
       +--rw pvid?                                  dot1qtypes:vlan-index-type
       +--rw default-priority?                      dot1qtypes:priority-type
       +--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 pcp-selection?                          dot1qtypes:pcp-selection-type
        +--rw pcp-decoding-table
        |  +--rw pcp-decoding-map* [pcp]
        |     +--rw pcp              pcp-selection-type
        |     +--rw priority-map* [priority-code-point]
        |        +--rw priority-code-point    priority-type
        |        +--rw priority?              priority-type
        |        +--rw drop-eligible?         boolean
        +--rw pcp-encoding-table
        |  +--rw pcp-encoding-map* [pcp]
        |     +--rw pcp              pcp-selection-type
        |     +--rw priority-map* [priority dei]
        |        +--rw priority               priority-type
        |        +--rw dei                    boolean
        |        +--rw priority-code-point?   priority-type
        +--rw use-dei?                                boolean
        +--rw drop-encoding?                          boolean
        +--rw service-access-priority-selection?      boolean
        +--rw service-access-priority
        |  +--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 traffic-class
        |  +--rw traffic-class-map* [priority]
        |     +--rw priority                    priority-type
        |     +--rw available-traffic-class* [num-traffic-class]
        |        +--rw num-traffic-class    uint8
        |        +--rw traffic-class?       traffic-class-type
        +--rw acceptable-frame?                       enumeration
        +--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
        +--rw admin-point-to-point?                   enumeration
        +--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-capabilties?                       bits
        +--ro external?                               boolean
        +--ro oper-point-to-point?                    boolean
        +--ro media-dependent-overhead?               uint8
        +--ro statistics
        |  +--ro delay-exceeded-discards?          yang:counter64
        |  +--ro mtu-exceeded-discards?            yang:counter64
        |  +--ro frame-rx?                         yang:counter64
        |  +--ro octets-rx?                        yang:counter64
        |  +--ro frame-tx?                         yang:counter64
        |  +--ro octets-tx?                        yang:counter64
        |  +--ro discard-inbound?                  yang:counter64
        |  +--ro forward-outbound?                 yang:counter64
        |  +--ro discard-lack-of-buffers?          yang:counter64
        |  +--ro discard-transit-delay-exceeded?   yang:counter64
        |  +--ro discard-on-error?                 yang:counter64
        |  +--ro discard-on-ingress-filtering?     yang:counter64 {ingress-
filtering}?
        +--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
```

### 48.6.4 Data scheme definition for the ieee802-dot1q-tpmr YANG module

```
module: ieee802-dot1q-tpmr
  augment /if:interfaces/if:interface/dot1q:bridge-port:
    +--rw managed-address?          boolean
    +--rw mac-status-propagation
       +--rw link-notify?         boolean
       +--rw link-notify-wait?    yang:timeticks
       +--rw link-notify-retry?   yang:timeticks
       +--rw mac-notify?          boolean
       +--rw mac-notify-time?     yang:timeticks
       +--rw mac-recover-time?    yang:timeticks
  augment /if:interfaces/if:interface/dot1q:bridge-port/dot1q:statistics:
    +--ro acks-tx?                      yang:counter64
    +--ro add-notificatons-tx?          yang:counter64
    +--ro loss-notification-tx?         yang:counter64
    +--ro loss-confirmation-tx?         yang:counter64
    +--ro acks-rx?                      yang:counter64
    +--ro add-notificatons-rx?          yang:counter64
    +--ro loss-notification-rx?         yang:counter64
    +--ro loss-confirmation-rx?         yang:counter64
    +--ro add-events?                   yang:counter64
    +--ro loss-events?                  yang:counter64
    +--ro mac-status-notifications?     yang:counter64
```

### 48.6.5 Data scheme definition for the ieee802-dot1q-vlan-bridge YANG module

N/A

### 48.6.6 Data scheme definition for the ieee802-dot1q-pb YANG module

```
module: ieee802-dot1q-pb
  augment /if:interfaces/if:interface/dot1q:bridge-port:
    +--rw svid?                              dot1qtypes:vlanid
    +--rw cvid-registration* [cvid]
    |  +--rw cvid            dot1qtypes:vlanid
    |  +--rw svid?           dot1qtypes:vlanid
    |  +--rw untagged-pep?   boolean
    |  +--rw untagged-cep?   boolean
    +--rw service-priority-regeneration* [svid]
    |  +--rw svid                       dot1qtypes: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              dot1qtypes:vlanid
       +--rw internal-port-number?      dot1qtypes:port-number-type
       +--rw internal-svid?             dot1qtypes:vlanid
       +--rw internal-interface-type?   enumeration
```

### 48.6.7 Data scheme definition for the ieee802-dot1q-stream-filters-gates YANG module

```
module: ieee802-dot1q-stream-filters-gates
  augment /dot1q:bridges/dot1q:bridge/dot1q:component:
    +--rw stream-filters
    |  +--rw stream-filter-instance-table* [stream-filter-instance-id]
    |  |  +--rw stream-filter-instance-id    uint32
    |  |  +--rw (stream-handle-spec)?
```

```
     |  |  |  +--:(wildcard)
     |  |  |  |  +--rw wildcard?              empty
     |  |  |  +--:(stream-handle)
     |  |  |     +--rw stream-handle          uint32
     |  |  +--rw priority-spec                ipv-type
     |  |  +--rw stream-gate-ref              stream-gate-ref
     |  |  +--rw filter-specification-list* [index]
     |  |     +--rw index                                                  uint8
     |  |     +--rw (filter-specification)?
     |  |        +--:(maximum-sdu-size)
     |  |           +--rw maximum-sdu-size                                 uint32
     |  |           +--rw stream-blocked-due-to-oversize-frame-enabled?    boolean
     |  |           +--ro stream-blocked-due-to-oversize-frame?            boolean
     |  +--ro max-stream-filter-instances?    uint32
     +--rw stream-gates
        +--rw stream-gate-instance-table* [stream-gate-instance-id]
        |  +--rw stream-gate-instance-id    uint32
        |  +--rw gate-enable?               boolean
        |  +--rw admin-gate-states?         gate-state-value-type
        |  +--rw admin-ipv?                 ipv-type
        |
        +--ro max-stream-gate-instances?    uint32
```

### 48.6.8 Data scheme definition for the ieee802-dot1q-ats YANG module

```
module: ieee802-dot1q-ats
  augment /dot1q:bridges/dot1q:bridge/dot1q:component
          /sfsg:stream-filters/sfsg:stream-filter-instance-table
          /sfsg:filter-specification-list/sfsg:filter-specification:
    +--:(scheduler-ref)
       +--rw scheduler-ref    ats:scheduler-ref-type
  augment /if:interfaces/if:interface/dot1q:bridge-port:
    +--rw ats-port-parameters
       +--ro discarded-frames-count?   yang:counter64
  augment /dot1q:bridges/dot1q:bridge/dot1q:component:
    +--rw schedulers
    |  +--rw scheduler-instance-table* [scheduler-instance-id]
    |  |  +--rw scheduler-instance-id         uint32
    |  |  +--rw committed-information-rate    uint64
    |  |  +--rw committed-burst-size          uint32
    |  |  +--rw scheduler-group-ref           ats:scheduler-group-ref-type
    |  +--ro max-scheduler-instances?    uint32
    +--rw scheduler-groups
       +--rw scheduler-group-instance-table* [scheduler-group-instance-id]
       |  +--rw scheduler-group-instance-id    uint32
       |  +--rw max-residence-time             uint32
       +--ro max-scheduler-group-instances?      uint32
       +--rw scheduler-timing-characteristics
                 +--ro  scheduler-timing-characteristics-table*  [reception-port
transmission-port]
            +--ro reception-port                   dot1qtypes:port-number-type
            +--ro transmission-port                dot1qtypes:port-number-type
            +--ro clock-offset-variation-max       uint32
            +--ro clock-rate-deviation-max         uint32
            +--ro arrival-recognition-delay-max    uint32
            +--ro processing-delay-min             uint32
            +--ro processing-delay-max             uint32
```

## 48.7 YANG modules

<<Editor's Note: Level 3 subclause names are symmetrically phrased to those for the MIB (clause 17.7 in IEEE Std 802.1Q-2018>>

### 48.7.1 Definitions for the ieee802-types YANG module

```
module ieee802-types {
  namespace urn:ieee:std:802.1Q:yang:ieee802-types;
  prefix ieee;
  organization
    "IEEE 802.1 Working Group";
  contact
   "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-L@ieee.org

    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
            P.O. Box 1331
            Piscataway
            NJ 08854
            USA

    E-mail: STDS-802-1-L@IEEE.ORG";
  description
    "This module contains a collection of generally useful derived
    data types for IEEE YANG models.";
  revision 2018-03-07 {
    description
      "Published as part of IEEE Std 802.1Q-2018.
      Initial version.";
    reference
      "IEEE Std 802.1Q-2018, Bridges and Bridged Networks.";
  }

  typedef mac-address {
    type string {
      pattern "[0-9a-fA-F]{2}(-[0-9a-fA-F]{2}){5}";
    }
    description
      "The mac-address type represents a MAC address in the canonical
      format and hexadecimal format specified by IEEE Std 802. The
      hexidecimal representation uses uppercase characters.";
    reference
      "3.1 of IEEE Std 802-2014
      8.1 of IEEE Std 802-2014";
  }
}
```

### 48.7.2 Definitions for the ieee802-dot1q-types YANG module

```
module ieee802-dot1q-types {
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-types;
  prefix dot1q-types;
  import ietf-yang-types {
    prefix yang;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-L@ieee.org

    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
```

```
            P.O. Box 1331
            Piscataway
            NJ 08854
            USA

    E-mail: STDS-802-1-L@IEEE.ORG";
  description
    "Common types used within dot1Q-bridge modules.";
  revision 2018-03-07 {
    description
      "Published as part of IEEE Std 802.1Q-2018.
      Initial version.";
    reference
      "IEEE Std 802.1Q-2018, Bridges and Bridged Networks.";
  }

  identity dot1q-vlan-type {
    description
      "Base identity from which all 802.1Q VLAN tag types are derived
      from.";
  }
  identity c-vlan {
    base dot1q-vlan-type;
    description
      "An 802.1Q Customer VLAN, using the 81-00 EtherType";
    reference
      "5.5 of IEEE Std 802.1Q-2018";
  }
  identity s-vlan {
    base dot1q-vlan-type;
    description
      "An 802.1Q Service VLAN, using the 88-A8 EtherType originally
      introduced in 802.1ad, and incorporated into 802.1Q (2011)";
    reference
      "5.6 of IEEE Std 802.1Q-2018";
  }
  typedef name-type {
    type string {
      length "0..32";
    }
    description
      "A text string of up to 32 characters, of locally determined
      significance.";
  }
  typedef port-number-type {
    type uint32 {
      range "1..65535";
    }
    description
      "The port number of the Bridge port for which this entry
      contains Bridge management information.";
  }
  typedef priority-type {
    type uint8 {
      range "0..7";
    }
    description
      "A range of priorities from 0 to 7 (inclusive). The Priority
      Code Point (PCP) is a 3-bit field that refers to the class of
      service associated with an 802.1Q VLAN tagged frame. The field
      specifies a priority value between 0 and 7, these values can be
      used by quality of service (QoS) to prioritize different classes
      of traffic.";
  }
  typedef vid-range-type {
    type string {
      pattern
        "([1-9]"+
        "[0-9]{0,3}"+
        "(-[1-9][0-9]{0,3})?"+
        "(,[1-9][0-9]{0,3}(-[1-9][0-9]{0,3})?)*)";
    }
```

```
    description
      "A list of VLAN Ids, or non overlapping VLAN ranges, in
      ascending order, between 1 and 4094.

      This type is used to match an ordered list of VLAN Ids, or
      contiguous ranges of VLAN Ids. Valid VLAN Ids must be in the
      range 1 to 4094, and included in the list in non overlapping
      ascending order.

      For example: 1,10-100,250,500-1000";
  }
  typedef vlanid {
    type uint16 {
      range "1..4094";
    }
    description
      "The vlanid type uniquely identifies a VLAN. This is the 12-bit
      VLAN-ID used in the VLAN Tag header. The range is defined by the
      referenced specification. This type is in the value set and its
      semantics equivalent to the VlanId textual convention of the
      SMIv2.";
  }
  typedef vlan-index-type {
    type uint32 {
      range "1..4094 | 4096..4294967295";
    }
    description
      "A value used to index per-VLAN tables. Values of 0 and 4095 are
      not permitted. The range of valid VLAN indices. If the value is
      greater than 4095, then it represents a VLAN with scope local to
      the particular agent, i.e., one without a global VLAN-ID
      assigned to it. Such VLANs are outside the scope of IEEE 802.1Q,
      but it is convenient to be able to manage them in the same way
      using this YANG module.";
    reference
      "9.6 of IEEE Std 802.1Q-2018";
  }
  typedef mstid-type {
    type uint32 {
      range "1..4094";
    }
    description
      "In an MSTP Bridge, an MSTID, i.e., a value used to identify a
      spanning tree (or MST) instance";
    reference
      "13.8 of IEEE Std 802.1Q-2018";
  }
  typedef pcp-selection-type {
    type enumeration {
      enum 8P0D {
        description
          "8 priorities, 0 drop eligible";
      }
      enum 7P1D {
        description
          "7 priorities, 1 drop eligible";
      }
      enum 6P2D {
        description
          "6 priorities, 2 drop eligible";
      }
      enum 5P3D {
        description
          "5 priorities, 3 drop eligible";
      }
    }
    description
      "Priority Code Point selection types.";
    reference
      "12.6.2.5.3 of IEEE Std 802.1Q-2018
      6.9.3 of IEEE Std 802.1Q-2018";
  }
```

```
  typedef protocol-frame-format-type {
    type enumeration {
      enum Ethernet {
        description
          "Ethernet frame format";
      }
      enum rfc1042 {
        description
          "RFC 1042 frame format";
      }
      enum snap8021H {
        description
          "SNAP 802.1H frame format";
      }
      enum snapOther {
        description
          "Other SNAP frame format";
      }
      enum llcOther {
        description
          "Other LLC frame format";
      }
    }
    description
      "A value representing the frame format to be matched.";
    reference
      "12.10.1.7.1 of IEEE Std 802.1Q-2018";
  }
  typedef ethertype-type {
    type string {
      pattern "[0-9a-fA-F]{2}-[0-9a-fA-F]{2}";
    }
    description
      "The EtherType value represented in the canonical order defined
      by IEEE 802. The canonical representation uses uppercase
      characters.";
    reference
      "9.2 of IEEE Std 802-2014";
  }
  typedef dot1q-tag-type {
    type identityref {
      base dot1q-vlan-type;
    }
    description
      "Identifies a specific 802.1Q tag type";
    reference
      "IEEE Std 802.1Q-2018";
  }
  typedef traffic-class-type {
    type uint8 {
      range "0..7";
    }
    description
      "This is the numerical value associated with a traffic class in
      a Bridge. Larger values are associated with higher priority
      traffic classes.";
    reference
      "3.239 of IEEE Std 802.1Q-2018";
  }
  grouping dot1q-tag-classifier-grouping {
    description
      "A grouping which represents an 802.1Q VLAN, matching both the
      EtherType and a single VLAN Id.";
    leaf tag-type {
      type dot1q-tag-type;
      mandatory true;
      description
        "VLAN type";
    }
    leaf vlan-id {
      type vlanid;
      mandatory true;
```

```
      description
        "VLAN Id";
    }
  }
  grouping dot1q-tag-or-any-classifier-grouping {
    description
      "A grouping which represents an 802.1Q VLAN, matching both the
      EtherType and a single VLAN Id or 'any' to match on any VLAN Id.";
    leaf tag-type {
      type dot1q-tag-type;
      mandatory true;
      description
        "VLAN type";
    }
    leaf vlan-id {
      type union {
        type vlanid;
        type enumeration {
          enum any {
            value 4095;
            description
              "Matches 'any' VLAN in the range 1 to 4094 that is not
              matched by a more specific VLAN Id match";
          }
        }
      }
      mandatory true;
      description
        "VLAN Id or any";
    }
  }
  grouping dot1q-tag-ranges-classifier-grouping {
    description
      "A grouping which represents an 802.1Q VLAN that matches a range
      of VLAN Ids.";
    leaf tag-type {
      type dot1q-tag-type;
      mandatory true;
      description
        "VLAN type";
    }
    leaf vlan-ids {
      type vid-range-type;
      mandatory true;
      description
        "VLAN Ids";
    }
  }
  grouping dot1q-tag-ranges-or-any-classifier-grouping {
    description
      "A grouping which represents an 802.1Q VLAN, matching both the
      EtherType and a single VLAN Id, ordered list of ranges, or 'any'
      to match on any VLAN Id.";
    leaf tag-type {
      type dot1q-tag-type;
      mandatory true;
      description
        "VLAN type";
    }
    leaf vlan-id {
      type union {
        type vid-range-type;
        type enumeration {
          enum any {
            value 4095;
            description
              "Matches 'any' VLAN in the range 1 to 4094.";
          }
        }
      }
      mandatory true;
      description
```

```
        "VLAN Ids or any";
    }
  }
  grouping priority-regeneration-table-grouping {
    description
      "The priority regeneration table provides the ability to map
      incoming priority values on a per-Port basis, under management
      control.";
    reference
      "6.9.4 of IEEE Std 802.1Q-2018";
    leaf priority0 {
      type priority-type;
      default "0";
      description
        "Priority 0";
      reference
        "12.6.2.3 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
    }
    leaf priority1 {
      type priority-type;
      default "1";
      description
        "Priority 1";
      reference
        "12.6.2.3 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
    }
    leaf priority2 {
      type priority-type;
      default "2";
      description
        "Priority 2";
      reference
        "12.6.2.3 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
    }
    leaf priority3 {
      type priority-type;
      default "3";
      description
        "Priority 3";
      reference
        "12.6.2.3 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
    }
    leaf priority4 {
      type priority-type;
      default "4";
      description
        "Priority 4";
      reference
        "12.6.2.3 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
    }
    leaf priority5 {
      type priority-type;
      default "5";
      description
        "Priority 5";
      reference
        "12.6.2.3 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
    }
    leaf priority6 {
      type priority-type;
      default "6";
      description
        "Priority 6";
      reference
        "12.6.2.3 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
```

```
    }
    leaf priority7 {
      type priority-type;
      default "7";
      description
        "Priority 7";
      reference
        "12.6.2.3 of IEEE Std 802.1Q-2018
        6.9.4 of IEEE Std 802.1Q-2018";
    }
  }
  grouping pcp-decoding-table-grouping {
    description
      "The Priority Code Point decoding table enables the decoding of
      the priority and drop-eligible parameters from the PCP.";
    reference
      "6.9.3 of IEEE Std 802.1Q-2018";
    list pcp-decoding-map {
      key "pcp";
      description
        "This map associates the priority code point field found in
        the VLAN to a priority and drop eligible value based upon the
        priority code point selection type.";
      leaf pcp {
        type pcp-selection-type;
        description
          "The priority code point selection type.";
        reference
          "12.6.2.7 of IEEE Std 802.1Q-2018
          6.9.3 of IEEE Std 802.1Q-2018";
      }
      list priority-map {
        key "priority-code-point";
        description
          "This map associated a priority code point value to priority
          and drop eligible parameters.";
        leaf priority-code-point {
          type priority-type;
          description
            "Priority associated with the pcp.";
          reference
            "12.6.2.7 of IEEE Std 802.1Q-2018
            6.9.3 of IEEE Std 802.1Q-2018";
        }
        leaf priority {
          type priority-type;
          description
            "Priority associated with the pcp.";
          reference
            "12.6.2.7 of IEEE Std 802.1Q-2018
            6.9.3 of IEEE Std 802.1Q-2018";
        }
        leaf drop-eligible {
          type boolean;
          description
            "Drop eligible value for pcp";
          reference
            "12.6.2.7 of IEEE Std 802.1Q-2018
            6.9.3 of IEEE Std 802.1Q-2018";
        }
      }
    }
  }
  grouping pcp-encoding-table-grouping {
    description
      "The Priority Code Point encoding table encodes the priority and
      drop-eligible parameters in the PCP field of the VLAN tag.";
    reference
      "12.6.2.9 of IEEE Std 802.1Q-2018
      6.9.3 of IEEE Std 802.1Q-2018";
    list pcp-encoding-map {
      key "pcp";
```

```
      description
        "This map associated the priority and drop-eligible parameters
        to the priority used to encode the PCP of the VLAN based upon
        the priority code point selection type.";
      leaf pcp {
        type pcp-selection-type;
        description
          "The priority code point selection type.";
        reference
          "12.6.2.7 of IEEE Std 802.1Q-2018
          6.9.3 of IEEE Std 802.1Q-2018";
      }
      list priority-map {
        key "priority dei";
        description
          "This map associated the priority and drop-eligible
          parameters to the priority code point field of the VLAN tag.";
        leaf priority {
          type priority-type;
          description
            "Priority associated with the pcp.";
          reference
            "12.6.2.7 of IEEE Std 802.1Q-2018
            6.9.3 of IEEE Std 802.1Q-2018";
        }
        leaf dei {
          type boolean;
          description
            "The drop eligible value.";
          reference
            "12.6.2 of IEEE Std 802.1Q-2018
            8.6.6 of IEEE Std 802.1Q-2018";
        }
        leaf priority-code-point {
          type priority-type;
          description
            "PCP value for priority when DEI value";
          reference
            "12.6.2.9 of IEEE Std 802.1Q-2018
            6.9.3 of IEEE Std 802.1Q-2018";
        }
      }
    }
  }
  grouping service-access-priority-table-grouping {
    description
      "The Service Access Priority Table associates a received
      priority with a serice access priority.";
    reference
      "12.6.2.17 of IEEE Std 802.1Q-2018
      6.13.1 of IEEE Std 802.1Q-2018";
    leaf priority0 {
      type priority-type;
      default "0";
      description
        "Service access priority value for priority 0";
      reference
        "12.6.2.17 of IEEE Std 802.1Q-2018
        6.13.1 of IEEE Std 802.1Q-2018";
    }
    leaf priority1 {
      type priority-type;
      default "1";
      description
        "Service access priority value for priority 1";
      reference
        "12.6.2.17 of IEEE Std 802.1Q-2018
        6.13.1 of IEEE Std 802.1Q-2018";
    }
    leaf priority2 {
      type priority-type;
      default "2";
```

```
      description
        "Service access priority value for priority 2";
      reference
        "12.6.2.17 of IEEE Std 802.1Q-2018
        6.13.1 of IEEE Std 802.1Q-2018";
    }
    leaf priority3 {
      type priority-type;
      default "3";
      description
        "Service access priority value for priority 3";
      reference
        "12.6.2.17 of IEEE Std 802.1Q-2018
        6.13.1 of IEEE Std 802.1Q-2018";
    }
    leaf priority4 {
      type priority-type;
      default "4";
      description
        "Service access priority value for priority 4";
      reference
        "12.6.2.17 of IEEE Std 802.1Q-2018
        6.13.1 of IEEE Std 802.1Q-2018";
    }
    leaf priority5 {
      type priority-type;
      default "5";
      description
        "Service access priority value for priority 5";
      reference
        "12.6.2.17 of IEEE Std 802.1Q-2018
        6.13.1 of IEEE Std 802.1Q-2018";
    }
    leaf priority6 {
      type priority-type;
      default "6";
      description
        "Service access priority value for priority 6";
      reference
        "12.6.2.17 of IEEE Std 802.1Q-2018
        6.13.1 of IEEE Std 802.1Q-2018";
    }
    leaf priority7 {
      type priority-type;
      default "7";
      description
        "Service access priority value for priority 7";
      reference
        "12.6.2.17 of IEEE Std 802.1Q-2018
        6.13.1 of IEEE Std 802.1Q-2018";
    }
  }
  grouping traffic-class-table-grouping {
    description
      "The Traffic Class Table models the operations that can be
      performed on, or inquire about, the current contents of the
      Traffic Class Table (8.6.6) for a given Port.";
    reference
      "12.6.3 of IEEE Std 802.1Q-2018
      8.6.6 of IEEE Std 802.1Q-2018";
    list traffic-class-map {
      key "priority";
      description
        "The priority index into the traffic class table.";
      leaf priority {
        type priority-type;
        description
          "The priority of the traffic class entry.";
        reference
          "8.6.6 of IEEE Std 802.1Q-2018";
      }
      list available-traffic-class {
```

```
        key "num-traffic-class";
        description
          "The traffic class index associated with a given priority
          within the traffic class table.";
        reference
          "8.6.6 of IEEE Std 802.1Q-2018";
        leaf num-traffic-class {
          type uint8 {
            range "1..8";
          }
          description
            "The available number of traffic classes.";
          reference
            "8.6.6 of IEEE Std 802.1Q-2018";
        }
        leaf traffic-class {
          type traffic-class-type;
          description
            "The traffic class index associated with a given traffic
            class entry.";
          reference
            "8.6.6 of IEEE Std 802.1Q-2018";
        }
      }
    }
  }
  grouping port-map-grouping {
    description
      "A set of control indicators, one for each Port. A Port Map,
      containing a control element for each outbound Port";
    reference
      "8.8.1 of IEEE Std 802.1Q-2018
      8.8.2 of IEEE Std 802.1Q-2018";
    list port-map {
      key "port-ref";
      description
        "The list of entries composing the port map.";
      leaf port-ref {
        type port-number-type;
        description
          "The interface port reference associated with this map.";
        reference
          "8.8.1 of IEEE Std 802.1Q-2018";
      }
      choice map-type {
        description
          "Type of port map";
        container static-filtering-entries {
          description
            "Static filtering entries attributes.";
          leaf control-element {
            type enumeration {
              enum forward {
                description
                  "Forwarded, independently of any dynamic filtering
                  information held by the FDB.";
              }
              enum filter {
                description
                  "Filtered, independently of any dynamic filtering
                  information.";
              }
              enum forward-filter {
                description
                  "Forwarded or filtered on the basis of dynamic
                  filtering information, or on the basis of the
                  default Group filtering behavior for the outbound
                  Port (8.8.6) if no dynamic filtering information is
                  present specifically for the MAC address.";
              }
            }
            description
```

```
            "containing a control element for each outbound Port,
            specifying that a frame with a destination MAC address,
            and in the case of VLAN Bridge components, VID that
            meets this specification.";
          reference
            "8.8.1 of IEEE Std 802.1Q-2018";
        }
        leaf connection-identifier {
          type port-number-type;
          description
            "A Port MAP may contain a connection identifier (8.8.12)
            for each outbound port. The connection identifier may be
            associated with the Bridge Port value maintained in a
            Dynamic Filtering Entry of the FDB for Bridge Ports.";
          reference
            "8.8.1 of IEEE Std 802.1Q-2018
            8.8.12 of IEEE Std 802.1Q-2018";
        }
      }
      container static-vlan-registration-entries {
        description
          "Static VLAN registration entries.";
        leaf registrar-admin-control {
          type enumeration {
            enum fixed-new-ignored {
              description
                "Registration Fixed (New ignored).";
            }
            enum fixed-new-propagated {
              description
                "Registration Fixed (New propagated.";
            }
            enum forbidden {
              description
                "Registration Forbidden.";
            }
            enum normal {
              description
                "Normal Registration.";
            }
          }
          description
            "The Registrar Administrative Control values for MVRP
            and MIRP for the VID.";
          reference
            "8.8.2 of IEEE Std 802.1Q-2018";
        }
        leaf vlan-transmitted {
          type enumeration {
            enum tagged {
              description
                "VLAN-tagged";
            }
            enum untagged {
              description
                "VLAN-untagged";
            }
          }
          description
            "Whether frames are to be VLAN-tagged or untagged when
            transmitted.";
          reference
            "8.8.2 of IEEE Std 802.1Q-2018";
        }
      }
      container mac-address-registration-entries {
        description
          "MAC address registration entries attributes.";
        leaf control-element {
          type enumeration {
            enum registered {
              description
```

```
                "Forwarded, independently of any dynamic filtering
                information held by the FDB.";
              }
              enum not-registered {
                description
                "Filtered, independently of any dynamic filtering
                information.";
              }
            }
            description
              "containing a control element for each outbound Port,
              specifying that a frame with a destination MAC address,
              and in the case of VLAN Bridge components, VID that
              meets this specification.";
            reference
              "8.8.4 of IEEE Std 802.1Q-2018";
          }
        }
        container dynamic-vlan-registration-entries {
          description
            "Dynamic VLAN registration entries attributes.";
          leaf control-element {
            type enumeration {
              enum registered {
                description
                "Forwarded, independently of any dynamic filtering
                information held by the FDB.";
              }
            }
            description
              "containing a control element for each outbound Port,
              specifying that a frame with a destination MAC address,
              and in the case of VLAN Bridge components, VID that
              meets this specification.";
            reference
              "8.8.5 of IEEE Std 802.1Q-2018";
          }
        }
        container dynamic-reservation-entries {
          description
            "Dynamic reservation entries attributes.";
          leaf control-element {
            type enumeration {
              enum forward {
                description
                "Forwarded, independently of any dynamic filtering
                information held by the FDB.";
              }
              enum filter {
                description
                "Filtered, independently of any dynamic filtering
                information.";
              }
            }
            description
              "Containing a control element for each outbound Port,
              specifying that a frame with a destination MAC address,
              and in the case of VLAN Bridge components, VID that
              meets this specification.";
            reference
              "8.8.7 of IEEE Std 802.1Q-2018";
          }
        }
        container dynamic-filtering-entries {
          description
            "Dynamic filtering entries attributes.";
          leaf control-element {
            type enumeration {
              enum forward {
                description
                "Forwarded, independently of any dynamic filtering
                information held by the FDB.";
```

```
            }
          }
          description
            "Containing a control element for each outbound Port,
            specifying that a frame with a destination MAC address,
            and in the case of VLAN Bridge components, VID that
            meets this specification.";
          reference
            "8.8.3 of IEEE Std 802.1Q-2018";
        }
      }
    }
  }
  grouping bridge-port-statistics-grouping {
    description
      "Grouping of bridge port statistics.";
    reference
      "12.6.1.1.3 of IEEE Std 802.1Q-2018";
    leaf delay-exceeded-discards {
      type yang:counter64;
      description
        "The number of frames discarded by this port due to excessive
        transit delay through the Bridge. It is incremented by both
        transparent and source route Bridges.";
      reference
        "12.6.1.1.3 of IEEE Std 802.1Q-2018
        8.6.6 of IEEE Std 802.1Q-2018";
    }
    leaf mtu-exceeded-discards {
      type yang:counter64;
      description
        "The number of frames discarded by this port due to an
        excessive size. It is incremented by both transparent and
        source route Bridges.";
      reference
        "12.6.1.1.3, item g) of IEEE Std 802.1Q-2018";
    }
    leaf frame-rx {
      type yang:counter64;
      description
        "The number of frames that have been received by this port
        from its segment. Note that a frame received on the interface
        corresponding to this port is only counted by this object if
        and only if it is for a protocol being processed by the local
        bridging function, including Bridge management frames.";
      reference
        "12.6.1.1.3 of IEEE Std 802.1Q-2018";
    }
    leaf octets-rx {
      type yang:counter64;
      description
        "The total number of octets in all valid frames received
        (including BPDUs, frames addressed to the Bridge as an end
        station, and frames that were submitted to the Forwarding
        Process).";
      reference
        "12.6.1.1.3 of IEEE Std 802.1Q-2018";
    }
    leaf frame-tx {
      type yang:counter64;
      description
        "The number of frames that have been transmitted by this port
        to its segment. Note that a frame transmitted on the interface
        corresponding to this port is only counted by this object if
        and only if it is for a protocol being processed by the local
        bridging function, including Bridge management frames.";
    }
    leaf octets-tx {
      type yang:counter64;
      description
        "The total number of octets that have been transmitted by this
```

```
          port to its segment.";
      }
      leaf discard-inbound {
        type yang:counter64;
        description
          "Count of received valid frames that were discarded (i.e.,
          filtered) by the Forwarding Process.";
        reference
          "12.6.1.1.3 of IEEE Std 802.1Q-2018";
      }
      leaf forward-outbound {
        type yang:counter64;
        description
          "The number of frames forwarded to the associated MAC Entity
          (8.5).";
        reference
          "12.6.1.1.3 of IEEE Std 802.1Q-2018";
      }
      leaf discard-lack-of-buffers {
        type yang:counter64;
        description
          "The count of frames that were to be transmitted through the
          associated Port but were discarded due to lack of buffers.";
        reference
          "12.6.1.1.3 of IEEE Std 802.1Q-2018";
      }
      leaf discard-transit-delay-exceeded {
        type yang:counter64;
        description
          "The number of frames discarded by this port due to excessive
          transit delay through the Bridge. It is incremented by both
          transparent and source route Bridges.";
        reference
          "12.6.1.1.3 of IEEE Std 802.1Q-2018";
      }
      leaf discard-on-error {
        type yang:counter64;
        description
          "The number of frames that were to be forwarded on the
          associated MAC but could not be transmitted (e.g., frame would
          be too large, 6.5.8).";
        reference
          "12.6.1.1.3 of IEEE Std 802.1Q-2018";
      }
    }
  }
```

### 48.7.3 Definitions for the ieee802-dot1q-bridge YANG module

```
module ieee802-dot1q-bridge {
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-bridge;
  prefix dot1q;
  import ieee802-types {
    prefix ieee;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-interfaces {
    prefix if;
  }
  import iana-if-type {
    prefix ianaif;
  }
  import ieee802-dot1q-types {
    prefix dot1qtypes;
  }
  organization
```

```
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-L@ieee.org

    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
            P.O. Box 1331
            Piscataway
            NJ 08854
            USA

    E-mail: STDS-802-1-L@IEEE.ORG";
  description
    "This YANG module describes the bridge configuration model for the
    following IEEE 802.1Q Bridges:
    1) Two Port MAC Relays
    2) Customer VLAN Bridges
    3) Provider Bridges.";
  revision 2018-03-07 {
    description
      "Published as part of IEEE Std 802.1Q-2018.
      Initial version.";
    reference
      "IEEE Std 802.1Q-2018, Bridges and Bridged Networks.";
  }

  feature ingress-filtering {
    description
      "Each Port may support an Enable Ingress Filtering parameter. A
      frame received on a Port that is not in the member set (8.8.10)
      associated with the frames VID shall be discarded if this
      parameter is set. The default value for this parameter is reset,
      i.e., Disable Ingress Filtering, for all Ports. Any Port that
      supports setting this parameter shall also support resetting it.
      The parameter may be configured by the management operations
      defined in Clause 12.";
    reference
      "8.6.2 of IEEE Std 802.1Q-2018";
  }
  feature extended-filtering-services {
    description
      "Extended Filtering Services support the filtering behavior
      required for regions of a network in which potential recipients
      of multicast frames exist, and where both the potential
      recipients of frames and the Bridges are able to support dynamic
      configuration of filtering information for group MAC addresses.
      In order to integrate this extended filtering behavior with the
      needs of regions of the network that support only Basic
      Filtering Services, Bridges that support Extended Filtering
      Services can be statically and dynamically configured to modify
      their filtering behavior on a per-group MAC address basis, and
      also on the basis of the overall filtering service provided by
      each outbound Port with regard to multicast frames. The latter
      capability permits configuration of the Ports default forwarding
      or filtering behavior with regard to group MAC addresses for
      which no specific static or dynamic filtering information has
      been configured.";
    reference
      "8.8.4 of IEEE Std 802.1Q-2018
      Clause 10 of IEEE Std 802.1Q-2018";
  }
  feature port-and-protocol-based-vlan {
    description
      "A VLAN-aware bridge component implementation in conformance to
      the provisions of this standard for Port-and-Protocol-based VLAN
      classification (5.4.1) shall 1) Support one or more of the
      following Protocol Classifications and Protocol Template
      formats: Ethernet, RFC_1042, SNAP_8021H, SNAP_Other, or
      LLC_Other (6.12); and may 2) Support configuration of the
```

```
      contents of the Protocol Group Database.";
    reference
      "5.4.1.2 of IEEE Std 802.1Q-2018";
  }
  feature flow-filtering {
    description
      "Flow filtering support enables Bridges to distinguish frames
      belonging to different client flows and to use this information
      in the forwarding process. Information related to client flows
      may be used at the boundary of an SPT Domain to generate a flow
      hash value. The flow hash, carried in an F-TAG, serves to
      distinguish frames belonging to different flows and can be used
      in the forwarding process to distribute frames over equal cost
      paths. This provides for finer granularity load spreading while
      maintaining frame order for each client flow.";
    reference
      "44.2 of IEEE Std 802.1Q-2018";
  }
  feature simple-bridge-port {
    description
      "A simple bridge port allows underlying (MAC) layers to share
      the same Interface as the Bridge Port.";
  }
  feature flexible-bridge-port {
    description
      "A flexible bridge port supports an Interface that is a Bridge
      Port to be a separate Interface from the underlying (MAC) layer.";
  }

  identity type-of-bridge {
    description
      "Represents the configured Bridge type.";
  }
  identity customer-vlan-bridge {
    base type-of-bridge;
    description
      "Base identity for a Customer VLAN Bridge.";
  }
  identity provider-bridge {
    base type-of-bridge;
    description
      "Base identity for a Provider Bridge (PB).";
  }
  identity provider-edge-bridge {
    base type-of-bridge;
    description
      "Base identity for a Provider Edge Bridge (PEB).";
  }
  identity two-port-mac-relay-bridge {
    base type-of-bridge;
    description
      "Base identity for a Two Port MAC Relay (TPMR).";
  }
  identity type-of-component {
    description
      "Represents the type of Component.";
  }
  identity c-vlan-component {
    base type-of-component;
    description
      "Base identity for a C-VLAN component.";
  }
  identity s-vlan-component {
    base type-of-component;
    description
      "Base identity for a S-VLAN component.";
  }
  identity d-bridge-component {
    base type-of-component;
    description
      "Base identity for a VLAN unaware component.";
  }
```

```
  identity edge-relay-component {
    base type-of-component;
    description
      "Base identity for an EVB station ER component.";
  }
  identity type-of-port {
    description
      "Represents the type of Bridge port.";
  }
  identity c-vlan-bridge-port {
    base type-of-port;
    description
      "Indicates the port can be a C-TAG aware port of an enterprise
      VLAN aware Bridge.";
  }
  identity provider-network-port {
    base type-of-port;
    description
      "Indicates the port can be an S-TAG aware port of a Provider
      Bridge or Backbone Edge Bridge used for connections within a PBN
      (Provider Bridged Network) or PBBN (Provider Backbone Bridged
      Network).";
  }
  identity customer-network-port {
    base type-of-port;
    description
      "Indicates the port can be an S-TAG aware port of a Provider
      Bridge or Backbone Edge Bridge used for connections to the
      exterior of a PBN (Provider Bridged Network) or PBBN (Provider
      Backbone Bridged Network).";
  }
  identity customer-edge-port {
    base type-of-port;
    description
      "Indicates the port can be a C-TAG aware port of a Provider
      Bridge used for connections to the exterior of a PBN (Provider
      Bridged Network) or PBBN (Provider Backbone Bridged Network).";
  }
  identity d-bridge-port {
    base type-of-port;
    description
      "Indicates the port can be a VLAN-unaware member of an 802.1Q
      Bridge.";
  }
  identity remote-customer-access-port {
    base type-of-port;
    description
      "Indicates the port can be an S-TAG aware port of a Provider
      Bridge capable of providing Remote Customer Service Interfaces.";
  }
  identity bridge-interface {
    description
      "Generic interface property that represents any interface that
      can be associated with an IEEE 802.1Q compliant Bridge
      component. Any new Interface types would derive from this
      identity to automatically pick up Bridge related configuration
      or operational data.";
  }

  container bridges {
    description
      "Contains the Bridge(s) configuration information.";
    list bridge {
      key "name";
      unique "address";
      description
        "Provides configuration data in support of the Bridge
        Configuration resources. There is a single bridge data node
        per Bridge.";
      leaf name {
        type dot1qtypes:name-type;
        description
```

```
        "A text string associated with the Bridge, of locally
        determined significance.";
      reference
        "12.4 of IEEE Std 802.1Q-2018";
    }
    leaf address {
      type ieee:mac-address;
      mandatory true;
      description
        "The MAC address for the Bridge from which the Bridge
        Identifiers used by the STP, RSTP, and MSTP are derived.";
      reference
        "12.4 of IEEE Std 802.1Q-2018";
    }
    leaf bridge-type {
      type identityref {
        base type-of-bridge;
      }
      mandatory true;
      description
        "The type of Bridge.";
    }
    leaf ports {
      type uint16 {
        range "1..4095";
      }
      config false;
      description
        "The number of Bridge Ports (MAC Entities)";
      reference
        "12.4 of IEEE Std 802.1Q-2018";
    }
    leaf up-time {
      type yang:zero-based-counter32;
      units "seconds";
      config false;
      description
        "The count in seconds of the time elapsed since the Bridge
        was last reset or initialized.";
      reference
        "12.4 of IEEE Std 802.1Q-2018";
    }
    leaf components {
      type uint32;
      config false;
      description
        "The number of components associated with the Bridge.";
    }
    list component {
      key "name";
      description
        "The set of components associated with a given Bridge. For
        example, - A TPMR is associated with a single VLAN
        unaware component. - A Customer VLAN Bridge is associated
        with a single VLAN aware component. - A Provider Bridge is
        associated with a single S-VLAN component and zero or more
        C-VLAN components.";
      reference
        "12.3 of IEEE Std 802.1Q-2018";
      leaf name {
        type string;
        description
          "The name of the Component.";
      }
      leaf id {
        type uint32;
        description
          "Unique identifier for a particular Bridge component
          within the system.";
        reference
          "12.3, item l) of IEEE Std 802.1Q-2018";
      }
```

```
        leaf type {
          type identityref {
            base type-of-component;
          }
          mandatory true;
          description
            "The type of component used to classify a particular
            Bridge component within a Bridge system comprising
            multiple components.";
          reference
            "12.3, item m) of IEEE Std 802.1Q-2018";
        }
        leaf address {
          type ieee:mac-address;
          description
            "Unique EUI-48 Universally Administered MAC address
            assigned to a Bridge component.";
          reference
            "13.24 of IEEE Std 802.1Q-2018
            8.13.8 of IEEE Std 802.1Q-2018";
        }
        leaf traffic-class-enabled {
          type boolean;
          default "true";
          description
            "Indication of Traffic Classes enablement associated with
            the Bridge Component. A value of True indicates that
            Traffic Classes are enabled on this Bridge Component. A
            value of False indicates that the Bridge Component
            operates with a single priority level for all traffic.";
          reference
            "12.4.1.5.1 of IEEE Std 802.1Q-2018";
        }
        leaf ports {
          type uint16 {
            range "1..4095";
          }
          config false;
          description
            "The number of Bridge Ports associated with the Bridge
            Component.";
          reference
            "12.4.1.1.3, item c) of IEEE Std 802.1Q-2018";
        }
        leaf-list bridge-port {
          type if:interface-ref;
          config false;
          description
            "List of bridge-port references.";
        }
        container capabilities {
          config false;
          description
            "Array of Boolean values of the feature capabilities
            associated with a given Bridge Component.";
          reference
            "12.10.1.1.3, item b) of IEEE Std 802.1Q-2018
            12.4.1.5.2 of IEEE Std 802.1Q-2018";
          leaf extended-filtering {
            type boolean;
            default "false";
            description
              "Can perform filtering on individual multicast addresses
              controlled by MMRP.";
            reference
              "12.4.1.5.2 of IEEE Std 802.1Q-2018";
          }
          leaf traffic-classes {
            type boolean;
            default "false";
            description
              "Can map priority to multiple traffic classes.";
```

```
        reference
          "12.4.1.5.2 of IEEE Std 802.1Q-2018";
      }
      leaf static-entry-individual-port {
        type boolean;
        default "false";
        description
          "Static entries per port.";
        reference
          "12.4.1.5.2 of IEEE Std 802.1Q-2018";
      }
      leaf ivl-capable {
        type boolean;
        default "true";
        description
          "Independent VLAN Learning (IVL).";
        reference
          "12.4.1.5.2 of IEEE Std 802.1Q-2018";
      }
      leaf svl-capable {
        type boolean;
        default "false";
        description
          "Shared VLAN Learning (SVL).";
        reference
          "12.4.1.5.2 of IEEE Std 802.1Q-2018";
      }
      leaf hybrid-capable {
        type boolean;
        default "false";
        description
          "Both IVL and SVL simultaneously.";
        reference
          "12.4.1.5.2 of IEEE Std 802.1Q-2018";
      }
      leaf configurable-pvid-tagging {
        type boolean;
        default "false";
        description
          "Whether the implementation supports the ability to
          override the default PVID setting and its egress status
          (VLAN-tagged or Untagged) on each port.";
        reference
          "12.4.1.5.2 of IEEE Std 802.1Q-2018";
      }
      leaf local-vlan-capable {
        type boolean;
        default "false";
        description
          "Can support multiple local Bridges, outside the scope
          of 802.1Q defined VLANs.";
        reference
          "12.4.1.5.2 of IEEE Std 802.1Q-2018";
      }
    }
    container filtering-database {
      when "../../bridge-type != 'two-port-mac-relay-bridge'" {
        description
          "Applies to non TPMRs.";
      }
      description
        "Contains filtering information used by the Forwarding
        Process in deciding through which Ports of the Bridge
        frames should be forwarded.";
      reference
        "12.7 of IEEE Std 802.1Q-2018";
      leaf aging-time {
        type uint32 {
          range "10..10000000";
        }
        units "seconds";
        default "300";
```

```
      description
        "The timeout period in seconds for aging out
        dynamically-learned forwarding information.";
      reference
        "12.7 of IEEE Std 802.1Q-2018
        8.8.3 of IEEE Std 802.1Q-2018";
    }
    leaf size {
      type yang:gauge32;
      config false;
      description
        "The maximum number of entries that can be held in the
        FDB.";
      reference
        "12.7 of IEEE Std 802.1Q-2018";
    }
    leaf static-entries {
      type yang:gauge32;
      config false;
      description
        "The number of Static Filtering entries currently in the
        FDB.";
      reference
        "12.7 of IEEE Std 802.1Q-2018
        8.8.1 of IEEE Std 802.1Q-2018";
    }
    leaf dynamic-entries {
      type yang:gauge32;
      config false;
      description
        "The number of Dynamic Filtering entries currently in
        the FDB.";
      reference
        "12.7 of IEEE Std 802.1Q-2018
        8.8.3 of IEEE Std 802.1Q-2018";
    }
    leaf static-vlan-registration-entries {
      type yang:gauge32;
      config false;
      description
        "The number of Static VLAN Registration entries
        currently in the FDB.";
      reference
        "12.7 of IEEE Std 802.1Q-2018
        8.8.2 of IEEE Std 802.1Q-2018";
    }
    leaf dynamic-vlan-registration-entries {
      type yang:gauge32;
      config false;
      description
        "The number of Dynamic VLAN Registration entries
        currently in the FDB.";
      reference
        "12.7 of IEEE Std 802.1Q-2018
        8.8.5 of IEEE Std 802.1Q-2018";
    }
    leaf mac-address-registration-entries {
      if-feature "extended-filtering-services";
      type yang:gauge32;
      config false;
      description
        "The number of MAC Address Registration entries
        currently in the FDB.";
      reference
        "12.7 of IEEE Std 802.1Q-2018
        8.8.4 of IEEE Std 802.1Q-2018";
    }
    list filtering-entry {
      key "database-id vids address";
      description
        "Information for the entries associated with the
        Permanent Database.";
```

```
            leaf database-id {
              type uint32;
              description
                "The identity of this Filtering Database.";
              reference
                "12.7.7 of IEEE Std 802.1Q-2018";
            }
            leaf address {
              type ieee:mac-address;
              description
                "A MAC address (unicast, multicast, broadcast) for
                which the device has forwarding and/or filtering
                information.";
              reference
                "12.7.7 of IEEE Std 802.1Q-2018";
            }
            leaf vids {
              type dot1qtypes:vid-range-type;
              description
                "The set of VLAN identifiers to which this entry
                applies.";
              reference
                "12.7.7 of IEEE Std 802.1Q-2018";
            }
            leaf entry-type {
              type enumeration {
                enum static {
                  description
                    "Static entry type";
                }
                enum dynamic {
                  description
                    "Dynamic/learnt entry type";
                }
              }
              description
                "The type of filtering entry. Whether static or
                dynamic. Static entries can be created, deleted, and
                retrieved. However, dynamic entries can only be
                deleted or retrieved by the management entity.
                Consequently, a Bridge is not required to accept a
                command that can alter the dynamic entries except
                delete a dynamic entry.";
              reference
                "12.7.7 of IEEE Std 802.1Q-2018";
            }
            uses dot1qtypes:port-map-grouping;
            leaf status {
              type enumeration {
                enum other {
                  description
                    "None of the following. This may include the case
                    where some other object is being used to determine
                    if and how frames addressed to the value of the
                    corresponding instance of 'address' are being
                    forwarded.";
                }
                enum invalid {
                  description
                    "This entry is no longer valid (e.g., it was
                    learned but has since aged out), but has not yet
                    been flushed from the table.";
                }
                enum learned {
                  description
                    "The value of the corresponding instance of the
                    port node was learned and is being used.";
                }
                enum self {
                  description
                    "The value of the corresponding instance of the
                    address node representing one of the devices
```

```
                    address.";
                  }
                  enum mgmt {
                    description
                      "The value of the corresponding instance of
                      address node that is also the value of an existing
                      instance.";
                  }
                }
                config false;
                description
                  "The status of this entry.";
              }
            }
            list vlan-registration-entry {
              key "database-id vids";
              description
                "The VLAN Registration Entries models the operations
                that can be performed on a single VLAN Registration
                Entry in the FDB. The set of VLAN Registration Entries
                within the FDB changes under management control and also
                as a result of MVRP exchanges";
              reference
                "12.7.5 of IEEE Std 802.1Q-2018";
              leaf database-id {
                type uint32;
                description
                  "The identity of this Filtering Database.";
                reference
                  "12.7.7 of IEEE Std 802.1Q-2018";
              }
              leaf vids {
                type dot1qtypes:vid-range-type;
                description
                  "The set of VLAN identifiers to which this entry
                  applies.";
                reference
                  "12.7.7 of IEEE Std 802.1Q-2018";
              }
              leaf entry-type {
                type enumeration {
                  enum static {
                    description
                      "Static entry type";
                  }
                  enum dynamic {
                    description
                      "Dynamic/learnt entry type";
                  }
                }
                description
                  "The type of filtering entry. Whether static or
                  dynamic. Static entries can be created, deleted, and
                  retrieved. However, dynamic entries can only be
                  deleted or retrieved by the management entity.
                  Consequently, a Bridge is not required to accept a
                  command that can alter the dynamic entries except
                  delete a dynamic entry.";
                reference
                  "12.7.7 of IEEE Std 802.1Q-2018";
              }
              uses dot1qtypes:port-map-grouping;
            }
          }
          container permanent-database {
            description
              "The Permanent Database container models the operations
              that can be performed on, or affect, the Permanent
              Database. There is a single Permanent Database per FDB.";
            leaf size {
              type yang:gauge32;
              config false;
```

```
        description
          "The maximum number of entries that can be held in the
          FDB.";
        reference
          "12.7.6 of IEEE Std 802.1Q-2018";
      }
      leaf static-entries {
        type yang:gauge32;
        config false;
        description
          "The number of Static Filtering entries currently in the
          FDB.";
        reference
          "12.7.6 of IEEE Std 802.1Q-2018";
      }
      leaf static-vlan-registration-entries {
        type yang:gauge32;
        config false;
        description
          "The number of Static VLAN Registration entries
          currently in the FDB.";
        reference
          "12.7.6 of IEEE Std 802.1Q-2018";
      }
      list filtering-entry {
        key "database-id vids address";
        description
          "Information for the entries associated with the
          Permanent Database.";
        leaf database-id {
          type uint32;
          description
            "The identity of this Filtering Database.";
          reference
            "12.7.7 of IEEE Std 802.1Q-2018";
        }
        leaf address {
          type ieee:mac-address;
          description
            "A MAC address (unicast, multicast, broadcast) for
            which the device has forwarding and/or filtering
            information.";
          reference
            "12.7.7 of IEEE Std 802.1Q-2018";
        }
        leaf vids {
          type dot1qtypes:vid-range-type;
          description
            "The set of VLAN identifiers to which this entry
            applies.";
          reference
            "12.7.7 of IEEE Std 802.1Q-2018";
        }
        leaf status {
          type enumeration {
            enum other {
              description
                "None of the following. This may include the case
                where some other object is being used to determine
                if and how frames addressed to the value of the
                corresponding instance of 'address' are being
                forwarded.";
            }
            enum invalid {
              description
                "This entry is no longer valid (e.g., it was
                learned but has since aged out), but has not yet
                been flushed from the table.";
            }
            enum learned {
              description
                "The value of the corresponding instance of the
```

```
                    port node was learned and is being used.";
                }
                enum self {
                  description
                    "The value of the corresponding instance of the
                    address node representing one of the devices
                    address.";
                }
                enum mgmt {
                  description
                    "The value of the corresponding instance of
                    address node that is also the value of an existing
                    instance.";
                }
              }
              config false;
              description
                "The status of this entry.";
            }
            uses dot1qtypes:port-map-grouping;
          }
        }
        container bridge-vlan {
          when "../../bridge-type != 'two-port-mac-relay-bridge'" {
            description
              "Applies to non TPMRs.";
          }
          description
            "The Bridge VLAN container models configuration
            information that modify, or inquire about, the overall
            configuration of the Bridges VLAN resources. There is a
            single Bridge VLAN Configuration managed object per
            Bridge.";
          reference
            "12.10 of IEEE Std 802.1Q-2018";
          leaf version {
            type uint16;
            config false;
            description
              "The version number supported.";
            reference
              "12.10.1.3 of IEEE Std 802.1Q-2018";
          }
          leaf max-vids {
            type uint16;
            config false;
            description
              "The maximum number of VIDs supported.";
            reference
              "12.10.1.3 of IEEE Std 802.1Q-2018";
          }
          leaf override-default-pvid {
            type boolean;
            default "false";
            config false;
            description
              "Indicates if the default PVID can be overridden, and
              its egress status (VLAN-tagged or untagged) on each
              port.";
            reference
              "12.10.1.3 of IEEE Std 802.1Q-2018";
          }
          leaf protocol-template {
            if-feature "port-and-protocol-based-vlan";
            type dot1qtypes:protocol-frame-format-type;
            config false;
            description
              "The data-link encapsulation format or the
              detagged_frame_type in a Protocol Template";
            reference
              "12.10.1.7 of IEEE Std 802.1Q-2018";
          }
```

```
        leaf max-msti {
          type uint16;
          config false;
          description
            "The maximum number of MSTIs supported within an MST
            region (i.e., the number of spanning tree instances that
            can be supported in addition to the CIST), for MST
            Bridges. For SST Bridges, this parameter may be either
            omitted or reported as 0.";
          reference
            "12.10.1.7 of IEEE Std 802.1Q-2018";
        }
        list vlan {
          key "vid";
          description
            "List of VLAN related configuration nodes associated
            with the Bridge.";
          reference
            "12.10.2 of IEEE Std 802.1Q-2018";
          leaf vid {
            type dot1qtypes:vlan-index-type;
            description
              "The VLAN identifier to which this entry applies.";
            reference
              "12.10.2 of IEEE Std 802.1Q-2018";
          }
          leaf name {
            type dot1qtypes:name-type;
            description
              "A text string of up to 32 characters of locally
              determined significance.";
            reference
              "12.10.2 of IEEE Std 802.1Q-2018";
          }
          leaf-list untagged-ports {
            type if:interface-ref;
            config false;
            description
              "The set of ports in the untagged set for this VID.";
            reference
              "12.10.2.1.3 of IEEE Std 802.1Q-2018
              8.8.2 of IEEE Std 802.1Q-2018";
          }
          leaf-list egress-ports {
            type if:interface-ref;
            config false;
            description
              "The set of egress ports in the member set for this
              VID.";
            reference
              "12.10.2.1.3 of IEEE Std 802.1Q-2018
              8.8.10 of IEEE Std 802.1Q-2018";
          }
        }
        list protocol-group-database {
          if-feature "port-and-protocol-based-vlan";
          key "db-index";
          description
            "List of the protocol group database entries.";
          reference
            "12.10.1.7 of IEEE Std 802.1Q-2018
            6.12.3 of IEEE Std 802.1Q-2018";
          leaf db-index {
            type uint16;
            description
              "The protocol group database index.";
          }
          leaf frame-format-type {
            type dot1qtypes:protocol-frame-format-type;
            description
              "The data-link encapsulation format or the
              detagged_frame_type in a Protocol Template";
```

```
      reference
        "12.10.1.7 of IEEE Std 802.1Q-2018";
    }
    choice frame-format {
      description
        "The identification of the protocol above the
        data-link layer in a Protocol Template. Depending on
        the frame type, the octet string will have one of the
        following values: - For ethernet, rfc1042 and
        snap8021H, this is the 16-bit (2-octet) IEEE 802
        Clause 9.3 EtherType field. - For snapOther, this is
        the 40-bit (5-octet) PID. - For llcOther, this is the
        2-octet IEEE 802.2 Link Service Access Point (LSAP)
        pair: first octet for Destination Service Access Point
        (DSAP) and second octet for Source Service Access
        Point (SSAP).";
      reference
        "12.10.1.7 of IEEE Std 802.1Q-2018";
      case ethernet-rfc1042-snap8021H {
        when
          "frame-format-type = 'Ethernet' or "+
          "frame-format-type = 'rfc1042' or "+
          "frame-format-type = 'snap8021H'" {
          description
            "Applies to Ethernet, RFC 1042, SNAP 8021H frame
            formats.";
        }
        description
          "Identifier used if Ethenet, RFC1042, or SNAP 8021H.";
        leaf ethertype {
          type dot1qtypes:ethertype-type;
          description
            "Format containing the 16-bit IEEE 802 EtherType
            field.";
          reference
            "9.3 of IEEE Std 802-2014";
        }
      }
      case snap-other {
        when "frame-format-type = 'snapOther'" {
          description
            "Applies to Snap Other frame formats.";
        }
        description
          "Identifier used if SNAP other.";
        leaf protocol-id {
          type string {
            pattern "[0-9a-fA-F]{2}(-[0-9a-fA-F]{2}){4}";
          }
          description
            "Format containing the 40-bit protocol identifier
            (PID). The canonical representation uses uppercase
            characters.";
          reference
            "12.10.1.7.1 of IEEE Std 802.1Q-2018";
        }
      }
      case llc-other {
        when "frame-format-type = 'llcOther'" {
          description
            "Applies to LLC Other frame formats";
        }
        description
          "Identifier used if LLC other.";
        container dsap-ssap-pairs {
          description
            "A pair of ISO/IEC 8802-2 DSAP and SSAP address
            field values, for matching frame formats of
            LLC_Other.";
          leaf llc-address {
            type string {
              pattern "[0-9a-fA-F]{2}-[0-9a-fA-F]{2}";
```

```
                    }
                    description
                      "A pair of ISO/IEC 8802-2 DSAP and SSAP address
                      field values, for matching frame formats of
                      LLC_Other. The canonical representation uses
                      uppercase characters.";
                    reference
                      "12.10.1.7.1 of IEEE Std 802.1Q-2018";
                  }
                }
              }
            }
            leaf group-id {
              type uint32;
              description
                "Designates a group of protocols in the Protocol Group
                Database.";
              reference
                "6.12.2 of IEEE Std 802.1Q-2018";
            }
          }
          list vid-to-fid-allocation {
            key "vids";
            description
              "This list allows inquiries about VID to FID
              allocations.";
            leaf vids {
              type dot1qtypes:vid-range-type;
              description
                "Range of VLAN identifiers.";
              reference
                "12.10.3 of IEEE Std 802.1Q-2018";
            }
            leaf fid {
              type uint32;
              config false;
              description
                "The Filtering Database used by a set of VIDs.";
              reference
                "12.10.3 of IEEE Std 802.1Q-2018";
            }
            leaf allocation-type {
              type enumeration {
                enum undefined {
                  description
                    "No allocation defined.";
                }
                enum fixed {
                  description
                    "A fixed allocation to FID is defined.";
                }
                enum dynamic {
                  description
                    "A dynamic allocation to FID is defined.";
                }
              }
              config false;
              description
                "The type of allocation used";
              reference
                "12.10.3 of IEEE Std 802.1Q-2018";
            }
          }
          list fid-to-vid-allocation {
            key "fid";
            description
              "The FID to VID allocations managed object models
              operations that inquire about FID to VID allocations.";
            leaf fid {
              type uint32;
              description
                "The Filtering Database used by a set of VIDs.";
```

```
            reference
              "12.10.3 of IEEE Std 802.1Q-2018";
          }
          leaf allocation-type {
            type enumeration {
              enum undefined {
                description
                  "No allocation defined.";
              }
              enum fixed {
                description
                  "A fixed allocation to FID is defined.";
              }
              enum dynamic {
                description
                  "A dynamic allocation to FID is defined.";
              }
            }
            config false;
            description
              "The type of allocation used";
            reference
              "12.10.3 of IEEE Std 802.1Q-2018";
          }
          leaf-list vid {
            type dot1qtypes:vlan-index-type;
            config false;
            description
              "The VLAN identifier to which this entry applies.";
            reference
              "12.7.7 of IEEE Std 802.1Q-2018";
          }
        }
        list vid-to-fid {
          key "vid";
          description
            "Fixed allocation of a VID to an FID. The underlying
            system will ensure that subsequent commands that make
            changes to the VID to FID mapping can override previous
            associations.";
          reference
            "12.10.3.4 of IEEE Std 802.1Q-2018
            12.10.3.5 of IEEE Std 802.1Q-2018";
          leaf vid {
            type dot1qtypes:vlan-index-type;
            description
              "A list of VLAN identifier associated with a given
              database identifier (i.e., FID).";
            reference
              "12.7.7 of IEEE Std 802.1Q-2018";
          }
          leaf fid {
            type uint32;
            description
              "The Filtering Database used by this VLAN";
            reference
              "12.10.3 of IEEE Std 802.1Q-2018";
          }
        }
      }
      container bridge-mst {
        when "../../bridge-type != 'two-port-mac-relay-bridge'" {
          description
            "Applies to non TPMRs.";
        }
        description
          "The Bridge MST container models configuration information
          that modify, or inquire about, the overall configuration
          of the Bridges MST resources.";
        reference
          "12.12 of IEEE Std 802.1Q-2018";
        leaf-list mstid {
```

```
            type dot1qtypes:mstid-type;
            description
              "The list of MSTID values that are currently supported
              by the Bridge";
          }
          list fid-to-mstid {
            key "fid";
            description
              "The FID to MSTID allocation table.";
            reference
              "12.12.2 of IEEE Std 802.1Q-2018";
            leaf fid {
              type uint32;
              description
                "The Filtering Database identifier.";
              reference
                "12.12.2 of IEEE Std 802.1Q-2018";
            }
            leaf mstid {
              type dot1qtypes:mstid-type;
              description
                "The MSTID to which the FID is to be allocated.";
              reference
                "12.12.2 of IEEE Std 802.1Q-2018";
            }
          }
          list fid-to-mstid-allocation {
            key "fids";
            description
              "The FID to MSTID allocation table";
            leaf fids {
              type dot1qtypes:vid-range-type;
              description
                "Range of FIDs.";
              reference
                "12.12.2 of IEEE Std 802.1Q-2018";
            }
            leaf mstid {
              type dot1qtypes:mstid-type;
              description
                "The MSTID to which the FID is allocated.";
              reference
                "12.12.2 of IEEE Std 802.1Q-2018";
            }
          }
        }
      }
    }
  }
  augment "/if:interfaces/if:interface" {
    when
      "if:type = 'ianaif:bridge' or if:type ="+
      "'ianaif:ethernetCsmacd' or if:type = 'ianaif:ieee8023adLag'"+
      "or if:type = 'ianaif:ilan'" {
      description
        "Applies when a Bridge interface.";
    }
    description
      "Augment the interface model with the Bridge Port";
    container bridge-port {
      description
        "Bridge Port is an extension of the IETF Interfaces model
        (RFC7223).";
      leaf component-name {
        type string;
        description
          "Used to reference configured Component node.";
      }
      leaf port-type {
        type identityref {
          base type-of-port;
        }
```

```
        description
          "The port type. Indicates the capabilities of this port.";
        reference
          "12.4.2.1 of IEEE Std 802.1Q-2018";
      }
      leaf pvid {
        when "../component-name != 'd-bridge-component'" {
          description
            "Applies to non TPMRs";
        }
        type dot1qtypes:vlan-index-type;
        default "1";
        description
          "The primary (default) VID assigned to a specific Bridge
          Port.";
        reference
          "12.10.1 of IEEE Std 802.1Q-2018
          5.4, item m) of IEEE Std 802.1Q-2018";
      }
      leaf default-priority {
        type dot1qtypes:priority-type;
        default "0";
        description
          "The default priority assigned to a specific Bridge Port.";
        reference
          "12.6.2 of IEEE Std 802.1Q-2018";
      }
      container priority-regeneration {
        description
          "The Priority Regeneration Table parameters associated with
          a specific Bridge Port. A list of Regenerated User
          Priorities for each received priority on each port of a
          Bridge. The regenerated priority value may be used to index
          the Traffic Class Table for each input port. This only has
          effect on media that support native priority. The default
          values for Regenerated User Priorities are the same as the
          User Priorities";
        reference
          "12.6.2 of IEEE Std 802.1Q-2018
          6.9.4 of IEEE Std 802.1Q-2018";
        uses dot1qtypes:priority-regeneration-table-grouping;
      }
      leaf pcp-selection {
        type dot1qtypes:pcp-selection-type;
        default "8P0D";
        description
          "The Priority Code Point selection assigned to a specific
          Bridge Port. This object identifies the rows in the PCP
          encoding and decoding tables that are used to remark frames
          on this port if this remarking is enabled";
        reference
          "12.6.2 of IEEE Std 802.1Q-2018
          6.9.3 of IEEE Std 802.1Q-2018";
      }
      container pcp-decoding-table {
        description
          "The Priority Code Point Decoding Table parameters
          associated with a specific Bridge Port.";
        uses dot1qtypes:pcp-decoding-table-grouping;
      }
      container pcp-encoding-table {
        description
          "The Priority Code Point Encoding Table parameters
          associated with a specific Bridge Port.";
        uses dot1qtypes:pcp-encoding-table-grouping;
      }
      leaf use-dei {
        type boolean;
        default "false";
        description
          "The Drop Eligible Indicator. If it is set to True, then the
          drop_eligible parameter is encoded in the DEI of transmitted
```

```
          frames, and the drop_eligible parameter shall be true(1) for
          a received frame if the DEI is set in the VLAN tag or the
          Priority Code Point Decoding Table indicates drop_eligible
          True for the received PCP value. If this parameter is False,
          the DEI shall be transmitted as zero and ignored on receipt.";
        reference
          "12.6.2 of IEEE Std 802.1Q-2018
          6.9.3 of IEEE Std 802.1Q-2018";
      }
      leaf drop-encoding {
        type boolean;
        default "false";
        description
          "The Drop Encoding parameter. If a Bridge supports encoding
          or decoding of drop_eligible from the PCP field of a VLAN
          tag (6.7.3) on any of its Ports, then it shall implement a
          Boolean parameter Require Drop Encoding on each of its Ports
          with default value False. If Require Drop Encoding is True
          and the Bridge Port cannot encode particular priorities with
          drop_eligible, then frames queued with those priorities and
          drop_eligible True shall be discarded and not transmitted.";
        reference
          "12.6.2 of IEEE Std 802.1Q-2018
          8.6.6 of IEEE Std 802.1Q-2018";
      }
      leaf service-access-priority-selection {
        type boolean;
        default "false";
        description
          "The Service Access Priority selection. Indication of
          whether the Service Access Priority Selection function is
          supported on the Customer Bridge Port to request priority
          handling of the frame from a Port-based service interface.";
        reference
          "12.6.2 of IEEE Std 802.1Q-2018
          6.13 of IEEE Std 802.1Q-2018";
      }
      container service-access-priority {
        description
          "The Service Access Priority table parameters. A table that
          contains information about the Service Access Priority
          Selection function for a Provider Bridge. The use of this
          table enables a mechanism for a Customer Bridge attached to
          a Provider Bridged Network to request priority handling of
          frames.";
        reference
          "12.6.2 of IEEE Std 802.1Q-2018
          6.13.1 of IEEE Std 802.1Q-2018";
        uses dot1qtypes:service-access-priority-table-grouping;
      }
      container traffic-class {
        description
          "The Traffic Class table parameters. A table mapping
          evaluated priority to Traffic Class, for forwarding by the
          Bridge";
        reference
          "12.6.3 of IEEE Std 802.1Q-2018
          8.6.6 of IEEE Std 802.1Q-2018";
        uses dot1qtypes:traffic-class-table-grouping;
      }
      leaf acceptable-frame {
        when "../component-name != 'd-bridge-component'" {
          description
            "Applies to non TPMRs";
        }
        type enumeration {
          enum admit-only-VLAN-tagged-frames {
            description
              "Admit only VLAN-tagged frames.";
          }
          enum admit-only-untagged-and-priority-tagged {
            description
```

```
            "Admit only untagged and priority-tagged frames.";
        }
        enum admit-all-frames {
          description
            "Admit all frames.";
        }
      }
      default "admit-all-frames";
      description
        "To configure the Acceptable Frame Types parameter
        associated with one or more Ports";
      reference
        "12.10.1.3 of IEEE Std 802.1Q-2018
        6.9 of IEEE Std 802.1Q-2018";
    }
    leaf enable-ingress-filtering {
      when "../component-name != 'd-bridge-component'" {
        description
          "Applies to non TPMRs";
      }
      type boolean;
      default "false";
      description
        "To enable the Ingress Filtering feature associated with one
        or more Ports.";
      reference
        "12.10.1.4 of IEEE Std 802.1Q-2018
        8.6.2 of IEEE Std 802.1Q-2018";
    }
    leaf enable-restricted-vlan-registration {
      when "../component-name != 'd-bridge-component'" {
        description
          "Applies to non TPMRs";
      }
      type boolean;
      default "false";
      description
        "To enable the Restricted VLAN Registration associated with
        one or more Ports.";
      reference
        "11.2.3.2.3 of IEEE Std 802.1Q-2018
        12.10.1.6 of IEEE Std 802.1Q-2018";
    }
    leaf enable-vid-translation-table {
      when "../component-name != 'd-bridge-component'" {
        description
          "Applies to non TPMRs";
      }
      type boolean;
      default "false";
      description
        "To enable VID Translation table associated with a Bridge
        Port. This is not applicable to Bridge Ports that do no
        support a VID Translation Table.";
      reference
        "12.10.1.8 of IEEE Std 802.1Q-2018
        6.9 of IEEE Std 802.1Q-2018";
    }
    leaf enable-egress-vid-translation-table {
      when "../component-name != 'd-bridge-component'" {
        description
          "Applies to non TPMRs";
      }
      type boolean;
      default "false";
      description
        "To enable Egress VID Translation table associated with a
        Bridge Port. This is not applicable to Ports that do not
        support an Egress VID Translation table.";
      reference
        "12.10.1.9 of IEEE Std 802.1Q-2018
        6.9 of IEEE Std 802.1Q-2018";
```

```
      }
      list protocol-group-vid-set {
        when "../component-name != 'd-bridge-component'" {
          description
            "Applies to non TPMRs";
        }
        if-feature "port-and-protocol-based-vlan";
        key "group-id";
        description
          "The list of VID values associated with the Protocol Group
          Identifier for this port.";
        reference
          "12.10.1.1.3 of IEEE Std 802.1Q-2018";
        leaf group-id {
          type uint32;
          description
            "The protocol group identifier";
          reference
            "12.10.1.7 of IEEE Std 802.1Q-2018";
        }
        leaf-list vid {
          type dot1qtypes:vlanid;
          description
            "The VLAN identifier to which this entry applies.";
          reference
            "12.10.2 of IEEE Std 802.1Q-2018";
        }
      }
      leaf admin-point-to-point {
        type enumeration {
          enum force-true {
            value 1;
            description
              "Indicates that this port should always be treated as if
              it is connected to a point-to-point link.";
          }
          enum force-false {
            value 2;
            description
              "Indicates that this port should be treated as having a
              shared media connection.";
          }
          enum auto {
            value 3;
            description
              "Indicates that this port is considered to have a
              point-to-point link if it is an Aggregator and all of
              its members are aggregatable, or if the MAC entity is
              configured for full duplex operation, either through
              auto-negotiation or by management means.";
          }
        }
        description
          "For a port running spanning tree, this object represents
          the administrative point-to-point status of the LAN segment
          attached to this port, using the enumeration values of IEEE
          Std 802.1AC. A value of forceTrue(1) indicates that this
          port should always be treated as if it is connected to a
          point-to-point link. A value of forceFalse(2) indicates that
          this port should be treated as having a shared media
          connection. A value of auto(3) indicates that this port is
          considered to have a point-to-point link if it is an
          Aggregator and all of its members are aggregatable, or if
          the MAC entity is configured for full duplex operation,
          either through auto-negotiation or by management means.
          Manipulating this object changes the underlying
          adminPointToPointMAC.";
        reference
          "12.4.2 of IEEE Std 802.1Q-2018
          6.8.2 of IEEE Std 802.1Q-2018";
      }
      leaf protocol-based-vlan-classification {
```

```
        when "../component-name != 'd-bridge-component'" {
          description
            "Applies to non TPMRs";
        }
        if-feature "port-and-protocol-based-vlan";
        type boolean;
        config false;
        description
          "A boolean indication indicating if Port-and-Protocol-based
          VLAN classification is supported on a given Port.";
        reference
          "5.4.1.2 of IEEE Std 802.1Q-2018";
      }
      leaf max-vid-set-entries {
        when "../component-name != 'd-bridge-component'" {
          description
            "Applies to non TPMRs";
        }
        if-feature "port-and-protocol-based-vlan";
        type uint16;
        config false;
        description
          "The maximum number of entries supported in the VID set on a
          given Port.";
        reference
          "12.10.1.1.3 of IEEE Std 802.1Q-2018";
      }
      leaf port-number {
        type dot1qtypes:port-number-type;
        config false;
        description
          "An integer that uniquely identifies a Bridge Port.";
        reference
          "12.3, item i) of IEEE Std 802.1Q-2018
          17.3.2.2 of IEEE Std 802.1Q-2018";
      }
      leaf address {
        type ieee:mac-address;
        config false;
        description
          "The specific MAC address of the individual MAC Entity
          associated with the Port.";
        reference
          "12.4.2 of IEEE Std 802.1Q-2018
          12.4.2.1.1.3, item a) of IEEE Std 802.1Q-2018";
      }
      leaf capabilities {
        type bits {
          bit tagging {
            position "0";
            description
              "Supports 802.1Q VLAN tagging of frames and MVRP.";
          }
          bit configurable-acceptable-frame-type {
            position "1";
            description
              "Allows modified values of acceptable frame types";
          }
          bit ingress-filtering {
            position "2";
            description
              "Supports the discarding of any frame received on a Port
              whose VLAN classification does not include that Port in
              its member set.";
          }
        }
        config false;
        description
          "The feature capabilities associated with port. Indicates
          the parts of IEEE 802.1Q that are optional on a per-port
          basis, that are implemented by this device, and that are
          manageable.";
```

```
        reference
          "12.10.1.1.3, item c) of IEEE Std 802.1Q-2018
          12.4.2 of IEEE Std 802.1Q-2018";
      }
      leaf type-capabilties {
        type bits {
          bit customer-vlan-port {
            position "0";
            description
              "Indicates the port can be a C-TAG aware port of an
              enterprise VLAN aware Bridge";
          }
          bit provider-network-port {
            position "1";
            description
              "Indicates the port can be an S-TAG aware port of a
              Provider Bridge or Backbone Edge Bridge used for
              connections within a PBN or PBBN.";
          }
          bit customer-network-port {
            position "2";
            description
              "Indicates the port can be an S-TAG aware port of a
              Provider Bridge or Backbone Edge Bridge used for
              connections to the exterior of a PBN or PBBN.";
          }
          bit customer-edge-port {
            position "3";
            description
              "Indicates the port can be a C-TAG aware port of a
              Provider Bridge used for connections to the exterior of
              a PBN or PBBN.";
          }
          bit customer-backbone-port {
            position "4";
            description
              "Indicates the port can be a I-TAG aware port of a
              Backbone Edge Bridge's B-component.";
          }
          bit virtual-instance-port {
            position "5";
            description
              "Indicates the port can be a virtual S-TAG aware port
              within a Backbone Edge Bridge's I-component which is
              responsible for handling S-tagged traffic for a specific
              backbone service instance.";
          }
          bit d-bridge-port {
            position "6";
            description
              "Indicates the port can be a VLAN-unaware member of an
              802.1Q Bridge.";
          }
          bit remote-customer-access-port {
            position "7";
            description
              "Indicates the port can be an S-TAG aware port of a
              Provider Bridge capable of providing Remote Customer
              Service Interfaces.";
          }
          bit station-facing-bridge-port {
            position "8";
            description
              "Indicates the station-facing Bridge Port in a EVB
              Bridge.";
          }
          bit uplink-access-port {
            position "9";
            description
              "Indicates the uplink access port in an EVB Bridge or
              EVB station.";
          }
```

```
            bit uplink-relay-port {
              position "10";
              description
                "Indicates the uplink relay port in an EVB station.";
            }
          }
          config false;
          description
            "The type of feature capabilities supported with port.
            Indicates the capabilities of this port.";
          reference
            "12.4.2 of IEEE Std 802.1Q-2018";
        }
        leaf external {
          type boolean;
          config false;
          description
            "A boolean indicating whether the port is external. A value
            of True means the port is external. A value of False means
            the port is internal.";
          reference
            "12.4.2 of IEEE Std 802.1Q-2018";
        }
        leaf oper-point-to-point {
          type boolean;
          config false;
          description
            "For a port running spanning tree, this object represents
            the operational point-to-point status of the LAN segment
            attached to this port. It indicates whether a port is
            considered to have a point-to-point connection.

            If admin-point-to-point is set to auto(2), then the value of
            oper-point-to-point is determined in accordance with the
            specific procedures defined for the MAC entity concerned, as
            defined in IEEE Std 802.1AC.

            The value is determined dynamically; that is, it is
            re-evaluated whenever the value of admin-point-to-point
            changes, and whenever the specific procedures defined for
            the MAC entity evaluate a change in its point-to-point
            status.";
          reference
            "IEEE Std 802.1AC
            12.4.2 of IEEE Std 802.1Q-2018";
        }
        container statistics {
          config false;
          description
            "Container of operational state node information associated
            with the bridge port.";
          uses dot1qtypes:bridge-port-statistics-grouping;
          leaf discard-on-ingress-filtering {
            when "../../component-name != 'd-bridge-component'" {
              description
                "Applies to non TPMRs";
            }
            if-feature "ingress-filtering";
            type yang:counter64;
            description
              "The number of frames that were discarded as a result of
              Ingress Filtering being enabled.

              Discontinuities in the value of this counter can occur at
              re-initialization of the management system, and at other
              times as indicated by the value of 'discontinuity-time'.";
            reference
              "12.6.1.1.3 of IEEE Std 802.1Q-2018";
          }
        }
        list vid-translations {
          when "../component-name != 'd-bridge-component'" {
```

```
        description
          "Applies to non TPMRs";
      }
      key "local-vid";
      description
        "To configure the VID Translation Table (6.9) associated
        with a Port. This object is not applicable to Ports that do
        not support a VID Translation Table. The default
        configuration of the table has the value of the Relay VID
        equal to the value of the Local VID. If no local VID is
        configured, then it is assumed that the relay VID is the
        same value as the local VID.

        If the port supports an Egress VID translation table, the
        VID Translation Configuration object configures the Local
        VID to Relay VID mapping on ingress only. If an Egress VID
        translation is not supported, the VID Translation
        Configuration object defines a single bidirectional mapping.
        In this case, the Bridge should not allow multiple keys
        ('local-vid') mapped to the same 'relay-vid' value.";
      leaf local-vid {
        type dot1qtypes:vlanid;
        description
          "The Local VID after translation received at the ISS or
          EISS.";
        reference
          "12.10.1.8 of IEEE Std 802.1Q-2018
          6.9 of IEEE Std 802.1Q-2018";
      }
      leaf relay-vid {
        type dot1qtypes:vlanid;
        description
          "The Relay VID received before translation received at ISS
          or EISS.";
        reference
          "12.10.1.8 of IEEE Std 802.1Q-2018
          6.9 of IEEE Std 802.1Q-2018";
      }
    }
    list egress-vid-translations {
      when "../component-name != 'd-bridge-component'" {
        description
          "Applies to non TPMRs";
      }
      key "relay-vid";
      description
        "To configure the Egress VID Translation Table (6.9)
        associated with a Port. This object is not applicable to
        Ports that do not support an Egress VID Translation Table.
        The default configuration of the table has the value of the
        Local VID equal to the value of the Relay VID. If no Relay
        VID is configured, then it is assumed that the local VID is
        the same value as the relay VID.";
      leaf relay-vid {
        type dot1qtypes:vlanid;
        description
          "The Relay VID received before translation received at ISS
          or EISS.";
        reference
          "12.10.1.9 of IEEE Std 802.1Q-2018
          6.9 of IEEE Std 802.1Q-2018";
      }
      leaf local-vid {
        type dot1qtypes:vlanid;
        description
          "The Local VID after translation received at the ISS or
          EISS.";
        reference
          "12.10.1.9 of IEEE Std 802.1Q-2018
          6.9 of IEEE Std 802.1Q-2018";
      }
    }
```

```
    }
  }
}
```

### 48.7.4 Definitions for the ieee802-dot1q-tpmr YANG module

```
module ieee802-dot1q-tpmr {
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-tpmr;
  prefix dot1q-tpmr;
  import ieee802-dot1q-bridge {
    prefix dot1q;
  }
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-interfaces {
    prefix if;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-L@ieee.org

    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
            P.O. Box 1331
            Piscataway
            NJ 08854
            USA

    E-mail: STDS-802-1-L@IEEE.ORG";
  description
    "This YANG module describes the bridge configuration model for the
    Two Port MAC Relays.";
  revision 2018-03-07 {
    description
      "Published as part of IEEE Std 802.1Q-2018.
      Initial version.";
    reference
      "IEEE Std 802.1Q-2018, Bridges and Bridged Networks.";
  }

  augment "/if:interfaces/if:interface/dot1q:bridge-port" {
    when "dot1q:port-type = 'dot1q:d-bridge-port'" {
      description
        "Applies to TPMRs ports";
    }
    description
      "Augment Interface model with TPMR port configuration
      specific nodes.";
    leaf managed-address {
      type boolean;
      default "true";
      description
        "A Boolean value, which is TRUE if the MAC address is the
        management address for the TPMR, and is otherwise FALSE.

        The TPMR management entity may make use of one or both Ports
        of a TPMR to transmit and receive management frames. However,
        the MAC address used by the TPMR management entity as the
        source MAC address in transmitted management frames (the
        management MAC address) is the individual MAC address
        associated with one of the Ports of the TPMR";
      reference
        "12.19.1.1.1.3 of IEEE Std 802.1Q-2018";
    }
    container mac-status-propagation {
      description
```

```
        "MAC status propagation configuration node parameters.";
      leaf link-notify {
        type boolean;
        default "true";
        description
          "The current value (Boolean) of LinkNotify (23.5.1) being
          used by the MSP state machines.";
        reference
          "12.19.4.1.1.3 of IEEE Std 802.1Q-2018
          12.19.4.1.2.2 of IEEE Std 802.1Q-2018";
      }
      leaf link-notify-wait {
        type yang:timeticks {
          range "20..100";
        }
        default "40";
        description
          "The current value, in centiseconds, of LinkNotifyWait
          (23.5.2) being used by the MSP state machines.";
        reference
          "12.19.4.1.1.3 of IEEE Std 802.1Q-2018
          12.19.4.1.2.2 of IEEE Std 802.1Q-2018";
      }
      leaf link-notify-retry {
        type yang:timeticks {
          range "10..100";
        }
        default "100";
        description
          "The current value, in centiseconds, of LinkNotifyRetry
          (23.5.3) being used by the MSP state machines.";
        reference
          "12.19.4.1.1.3 of IEEE Std 802.1Q-2018
          12.19.4.1.2.2 of IEEE Std 802.1Q-2018";
      }
      leaf mac-notify {
        type boolean;
        default "true";
        description
          "The current value (Boolean) of MACNotify (23.5.4) being
          used by the MSP state machines.";
        reference
          "12.19.4.1.1.3 of IEEE Std 802.1Q-2018
          12.19.4.1.2.2 of IEEE Std 802.1Q-2018";
      }
      leaf mac-notify-time {
        type yang:timeticks {
          range "1..50";
        }
        default "20";
        description
          "The current value, in centiseconds, of MACNotifyTime
          (23.5.5) being used by the MSP state machines.";
        reference
          "12.19.4.1.1.3 of IEEE Std 802.1Q-2018
          12.19.4.1.2.2 of IEEE Std 802.1Q-2018";
      }
      leaf mac-recover-time {
        type yang:timeticks {
          range "2..50";
        }
        default "10";
        description
          "The current value, in centiseconds, of MACRecoverTime
          (23.5.6) being used by the MSP state machines.";
        reference
          "12.19.4.1.1.3 of IEEE Std 802.1Q-2018
          12.19.4.1.2.2 of IEEE Std 802.1Q-2018";
      }
    }
  }
  augment
```

```
"/if:interfaces/if:interface/dot1q:bridge-port/dot1q:statistics" {
when "../dot1q:port-type = 'dot1q:d-bridge-port'" {
  description
    "Applies to TPMRs ports";
}
description
  "Augment Interface model with TPMR port operational state
  specific nodes.";
leaf acks-tx {
  type yang:counter64;
  config false;
  description
    "The number of acks transmitted (23.6.15) by the Ports
    Transmit Process as a consequence of txAck being set.

    Discontinuities in the value of this counter can occur at
    re-initialization of the management system, and at other times
    as indicated by the value of 'discontinuity-time'.";
  reference
    "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
}
leaf add-notificatons-tx {
  type yang:counter64;
  config false;
  description
    "The number of adds transmitted (23.6.16) by the Ports
    Transmit Process as a consequence of txAdd being set.

    Discontinuities in the value of this counter can occur at
    re-initialization of the management system, and at other times
    as indicated by the value of 'discontinuity-time'.";
  reference
    "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
}
leaf loss-notification-tx {
  type yang:counter64;
  config false;
  description
    "The number of losses transmitted (23.6.18) by the Ports
    Transmit Process as a consequence of txLoss being set.

    Discontinuities in the value of this counter can occur at
    re-initialization of the management system, and at other times
    as indicated by the value of 'discontinuity-time'.";
  reference
    "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
}
leaf loss-confirmation-tx {
  type yang:counter64;
  config false;
  description
    "The number of loss confirms transmitted (23.6.19) by the
    Ports Transmit Process as a consequence of txLossConfirm being
    set.

    Discontinuities in the value of this counter can occur at
    re-initialization of the management system, and at other times
    as indicated by the value of 'discontinuity-time'.";
  reference
    "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
}
leaf acks-rx {
  type yang:counter64;
  config false;
  description
    "The number of acks received (23.6.10) by the Ports Transmit
    Process.

    Discontinuities in the value of this counter can occur at
    re-initialization of the management system, and at other times
    as indicated by the value of 'discontinuity-time'.";
  reference
```

```
        "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
    }
    leaf add-notificatons-rx {
      type yang:counter64;
      config false;
      description
        "The number of adds received (23.6.11) by the Ports Receive
        Process.

        Discontinuities in the value of this counter can occur at
        re-initialization of the management system, and at other times
        as indicated by the value of 'discontinuity-time'.";
      reference
        "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
    }
    leaf loss-notification-rx {
      type yang:counter64;
      config false;
      description
        "The number of losses received (23.6.13) by the Ports Receive
        Process.

        Discontinuities in the value of this counter can occur at
        re-initialization of the management system, and at other times
        as indicated by the value of 'discontinuity-time'.";
      reference
        "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
    }
    leaf loss-confirmation-rx {
      type yang:counter64;
      config false;
      description
        "The number of loss confirms received (23.6.14) by the Ports
        Receive Process.

        Discontinuities in the value of this counter can occur at
        re-initialization of the management system, and at other times
        as indicated by the value of 'discontinuity-time'.";
      reference
        "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
    }
    leaf add-events {
      type yang:counter64;
      config false;
      description
        "The number of transitions to STM:ADD directly from STM:DOWN
        or STM:LOSS (23.8).

        Discontinuities in the value of this counter can occur at
        re-initialization of the management system, and at other times
        as indicated by the value of 'discontinuity-time'.";
      reference
        "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
    }
    leaf loss-events {
      type yang:counter64;
      config false;
      description
        "The number of transitions to STM:LOSS directly from STM:UP or
        STM:ADD (23.8).

        Discontinuities in the value of this counter can occur at
        re-initialization of the management system, and at other times
        as indicated by the value of 'discontinuity-time'.";
      reference
        "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
    }
    leaf mac-status-notifications {
      type yang:counter64;
      config false;
      description
        "The number of transitions to SNM:MAC_NOTIFICATION (23.9).
```

```
        Discontinuities in the value of this counter can occur at
        re-initialization of the management system, and at other times
        as indicated by the value of 'discontinuity-time'.";
      reference
        "12.19.4.1.3.3 of IEEE Std 802.1Q-2018";
    }
  }
}
```

### 48.7.5 Definitions for the ieee802-dot1q-vlan-bridge YANG module

```
module ieee802-dot1q-vlan-bridge {
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-vlan-bridge;
  prefix dot1q-vlan-bridge;
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-L@ieee.org

    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
            P.O. Box 1331
            Piscataway
            NJ 08854
            USA

    E-mail: STDS-802-1-L@IEEE.ORG";
  description
    "This YANG module describes the bridge configuration model for
    Customer VLAN Bridges.";
  revision 2018-03-07 {
    description
      "Published as part of IEEE Std 802.1Q-2018.
      Initial version.";
    reference
      "IEEE Std 802.1Q-2018, Bridges and Bridged Networks.";
  }
}
```

### 48.7.6 Definitions for the ieee802-dot1q-pb YANG module

```
module ieee802-dot1q-pb {
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-pb;
  prefix dot1q-pb;
  import ieee802-dot1q-bridge {
    prefix dot1q;
  }
  import ieee802-dot1q-types {
    prefix dot1qtypes;
  }
  import ietf-interfaces {
    prefix if;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-L@ieee.org

    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
            P.O. Box 1331
            Piscataway
            NJ 08854
```

```
            USA

    E-mail: STDS-802-1-L@IEEE.ORG";
  description
    "This YANG module describes the bridge configuration model for
    Provider Bridges.";
  revision 2018-03-07 {
    description
      "Published as part of IEEE Std 802.1Q-2018.
      Initial version.";
    reference
      "IEEE Std 802.1Q-2018, Bridges and Bridged Networks.";
  }

  augment "/if:interfaces/if:interface/dot1q:bridge-port" {
    description
      "Augment the interface model with 802.1Q Bridge Port
      configuration specific nodes.";
    leaf svid {
      type dot1qtypes:vlanid;
      description
        "Service VLAN identifier.";
      reference
        "12.13.2.1 of IEEE Std 802.1Q-2018";
    }
    list cvid-registration {
      when
        "../dot1q:component-name = 'dot1q:c-vlan-component' and "+
        "../dot1q:port-type = 'dot1q:customer-edge-port'" {
        description
          "Applies when the component associated with this interface
          is a C-VLAN component and the port-type is a customer edge
          port.";
      }
      key "cvid";
      description
        "The C-VID Registration Table, provides a mapping between a
        C-VID and the service instance represented by an S-VID
        selected for that C-VLAN. This table provides the equivalent
        functionality of
           1) Configuring the PVID of the internal CNP on the S-VLAN
              component
           2) Adding the corresponding PEP on the C-VLAN component to
              the member set of the C-VLAN
           3) Adding the PEP and/or CEP to the untagged set of the
              C-VLAN (if it is desired that frames forwarded to that
              port are transmitted untagged for this C-VLAN).";
      leaf cvid {
        type dot1qtypes:vlanid;
        description
          "Customer VLAN identifiers associated with this bridge port.";
        reference
          "12.13.2.1 of IEEE Std 802.1Q-2018";
      }
      leaf svid {
        type dot1qtypes:vlanid;
        description
          "Service VLAN identifier.";
        reference
          "12.13.2.1 of IEEE Std 802.1Q-2018";
      }
      leaf untagged-pep {
        type boolean;
        default "true";
        description
          "A boolean indicating frames for this C-VLAN should be
          forwarded untagged through the Provider Edge Port.";
        reference
          "12.13.2.1 of IEEE Std 802.1Q-2018";
      }
      leaf untagged-cep {
        type boolean;
```

```
        default "true";
        description
          "A boolean indicating frames for this C-VLAN should be
          forwarded untagged through the Customer Edge Port.";
        reference
          "12.13.2.1 of IEEE Std 802.1Q-2018";
      }
    }
    list service-priority-regeneration {
      when
        "../dot1q:component-name = 'dot1q:c-vlan-component' and "+
        "../dot1q:port-type = 'dot1q:customer-edge-port'" {
        description
          "Applies when the component associated with this interface
          is a C-VLAN component and the port-type is a customer edge
          port.";
      }
      key "svid";
      description
        "The Service Priority Regeneration Table, which provides the
        Priority Regeneration Table (12.6.2) for each internal CNP
        connected to the C-VLAN component associated with the CEP.";
      leaf svid {
        type dot1qtypes:vlanid;
        description
          "Service VLAN identifier.";
        reference
          "12.13.2.6 of IEEE Std 802.1Q-2018";
      }
      container priority-regeneration {
        description
          "Contains Service Priority Regeneration table nodal
          information.";
        reference
          "12.13.2.6 of IEEE Std 802.1Q-2018";
        uses dot1qtypes:priority-regeneration-table-grouping;
      }
    }
    list rcap-internal-interface {
      when
        "../dot1q:component-name = 'dot1q:s-vlan-component' and "+
        "../dot1q:port-type = 'dot1q:remote-customer-access-port'" {
        description
          "Applies when the component associated with this interface
          is a C-VLAN component and the port-type is a customer edge
          port.";
      }
      key "external-svid";
      description
        "Designating an external port as an RCAP automatically creates
        a Port-mapping S-VLAN component associated with that port.
        This Port-mapping S-VLAN component includes one internal PNP.";
      leaf external-svid {
        type dot1qtypes:vlanid;
        description
          "External Service VLAN identifier.";
        reference
          "12.13.3.2 of IEEE Std 802.1Q-2018";
      }
      leaf internal-port-number {
        type dot1qtypes:port-number-type;
        description
          "The number of the RCAP.";
        reference
          "12.13.3.2 of IEEE Std 802.1Q-2018";
      }
      leaf internal-svid {
        type dot1qtypes:vlanid;
        description
          "Internal Service VLAN Identifier (not applicable for a
          C-tagged RCSI).";
        reference
```

```
            "12.13.3.2 of IEEE Std 802.1Q-2018";
        }
        leaf internal-interface-type {
          type enumeration {
            enum port-based-rcsi {
              description
                "Port-based RCSI";
            }
            enum c-tagged-rcsi {
              description
                "C-tagged RCSI";
            }
            enum pnp {
              description
                "Provider Network Port";
            }
            enum discard {
              description
                "Discard (external S-VID is not associated with an
                internal port).";
            }
          }
          description
            "A value indicating the type of internal interface
            associated with the external S-VID.";
          reference
            "12.13.3.2 of IEEE Std 802.1Q-2018";
        }
      }
    }
  }
```

### 48.7.7 Definitions for the ieee802-dot1q-stream-filters-gates YANG module

```
module ieee802-dot1q-stream-filters-gates {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-stream-filters-gates;
  prefix sfsg;
  import ieee802-dot1q-bridge {
    prefix dot1q;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: https://1.ieee802.org/
    WG-EMail: stds-802-1@ieee.org
    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
            P.O. Box 1331
            Piscataway
            NJ 08855-1331
            USA

    E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG";
  description
    "This module provides management of 802.1Q bridge components that
    support Stream Filters and Stream Gates. NOTICE: This YANG module is
    part of an unapproved IEEE Standards Draft and is subject to change.";
  revision 2019-04-13 {
    description
      "Initial revision from IEEE P802.1Qcr.";
    reference
```

```
      "IEEE Std 802.1Qcr";
  }
  feature closed-gate-state {
    description
      "The bridge component supports gate state closed.";
    reference
      "IEEE Std 802.1Qcr";
  }

  /* Types and groupings */
  typedef ipv-type {
    type enumeration {
      enum zero {
        value 0;
        description
          "Priority 0";
      }
      enum one {
        value 1;
        description
          "Priority 1";
      }
      enum two {
        value 2;
        description
          "Priority 2";
      }
      enum three {
        value 3;
        description
          "Priority 3";
      }
      enum four {
        value 4;
        description
          "Priority 4";
      }
      enum five {
        value 5;
        description
          "Priority 5";
      }
      enum six {
        value 6;
        description
          "Priority 6";
      }
      enum seven {
        value 7;
        description
          "Priority 7";
      }

      /* NOTE: The mapping of the wildcard literal is other
       *       than in the MIB definition, where the wildcard
       *       value is mapped to -1.
       */
      enum wildcard {
        description
```

```
            "No Priority";
        }
      }
      description
        "An IPV can be either of the following:
           1) The null value. For a frame that passes through the gate, the
              priority value associated with the frame is used to determine
              the frame's traffic class, using the Traffic Class Table as
              specified in 8.6.6.
           2) An internal priority value. For a frame that passes through the
              gate, the IPV is used, in place of the priority value
              associated with the frame, to determine the frame's traffic
              class, using the Traffic Class Table as specified in 8.6.6.";
      reference
        "8.6.5.2 of IEEE Std 802.1Qcr";
    }
    typedef gate-state-value-type {
      type enumeration {
        enum open {
          description
            "Gate open";
        }
        enum closed {
          description
            "Gate closed";
        }
      }
      description
        "The StreamGateStatesValue indicates the desired gate state, open or
        closed, for the stream gate.";
      reference
        "12.31.3.2.1 of IEEE Std 802.1Qcr";
    }
    typedef stream-gate-ref {
      type leafref {
        path
          '/dot1q:bridges'+
          '/dot1q:bridge'+
          '/dot1q:component'+
          '/sfsg:stream-gates'+
          '/sfsg:stream-gate-instance-table'+
          '/sfsg:stream-gate-instance-id';
      }
      description
        "This type is used to refer to a stream gate instance.";
    }
    augment "/dot1q:bridges/dot1q:bridge/dot1q:component" {
      description
        "Augments the Bridge component with Stream Filters and Stream Gates.";
      container stream-filters {
        description
          "This container encapsulates all nodes related to Stream Filters.";
        reference
          "12.31.1 of IEEE Std 802.1Qcr
          12.31.2 of IEEE Std 802.1Qcr
          12.31.3 of IEEE Std 802.1Qcr";
        list stream-filter-instance-table {
          key "stream-filter-instance-id";
          description
```

```
        "There is one Stream Filter Instance Table per Bridge component.
        Each table row contains a set of parameters that defines a single
        Stream Filter (8.6.5.1), as detailed in Table 12-31. The table
        rows form an ordered list of filter instances, the order being
        determined by the StreamFilterInstance parameter. Tables can be
        created or removed dynamically in implementations that support
        dynamic configuration of Bridge components. Rows in the table can
        be created or removed dynamically in implementations that support
        dynamic configuration of stream filters. The value of the
        stream-handle-spec and priority-spec parameters associated with a
        received frame determine which stream filter is selected by the
        frame, and therefore what combination of filtering and policing
        actions is applied to the frame. If the stream-handle-spec and
        priority-spec parameters associated with a received frame match
        more than one stream filter, the stream filter that is selected
        is the one that appears earliest in the ordered list. If a
        received frame's stream-handle-spec and priority-spec does not
        match any of the stream filters in the table, the frame is
        processed as if Stream Filters and Stream Gates would not be
        supported.";
      reference
        "12.31.2 of IEEE Std 802.1Qcr";
      leaf stream-filter-instance-id {
        type uint32;
        mandatory true;
        description
          "An integer index value that determines the place of the stream
          filter in the ordered list of stream filter instances. The
          values are ordered according to their integer value; smaller
          values appear earlier in the ordered list.";
        reference
          "12.31.2.1 of IEEE Std 802.1Qcr";
      }
      choice stream-handle-spec {
        description
          "The stream_handle specification data type allows either of the
          following to be represented:
           a) A stream_handle value, represented as an integer.
           b) The wild card value, which matches any frame";
        reference
          "12.31.2.2 of IEEE Std 802.1Qcr";

        /* NOTE: The mapping of the wildcard literal is
         *       other than in the MIB definition, where
         *       the wildcard value is mapped to -1.
         */
        case wildcard {
          leaf wildcard {
            type empty;
            description
              "The stream handle specification represents a wild card
              value.";
          }
        }
        case stream-handle {
          leaf stream-handle {
            type uint32;
            mandatory true;
            description
```

```
            "The stream handle specification refers to a stream_handle
            value.";
        }
      }
    }
    leaf priority-spec {
      type ipv-type;
      mandatory true;
      description
        "The priority specification data type allows either of the
        following to be represented:
         a) A priority value, represented as an integer.
         b) The wild card value, which matches any priority.";
      reference
        "12.31.2.3 of IEEE Std 802.1Qcr";
    }
    leaf stream-gate-ref {
      type stream-gate-ref;
      mandatory true;
      description
        "The StreamGateInstance parameter identifies the stream gate
        (12.31.3) that is associated with the stream filter. The
        relationship between stream filters and stream gates is many to
        one; a given stream filter can be associated with only one
        stream gate, but there can be multiple stream filters
        associated with a given stream gate.";
      reference
        "12.31.2.4 of IEEE Std 802.1Qcr";
    }
    list filter-specification-list {
      key "index";
      description
        "The filter specification list contains one or more filter
        specifications that are assigned with this stream filter.";
      reference
        "12.31.2.5 of IEEE Std 802.1Qcr";
      leaf index {
        type uint8;
        description
          "The index of this filter specification.";
      }
      choice filter-specification {
        description
          "The filter specification type and its parameters.";
        reference
          "12.31.2.5 of IEEE Std 802.1Qcr
          8.6.5.1 of IEEE Std 802.1Qcr
          8.6.5.3 of IEEE Std 802.1Qcr";
        case maximum-sdu-size {
          description
            "Maximum SDU size filter";
          leaf maximum-sdu-size {
            type uint32;
            mandatory true;
            description
              "The allowed maximum SDU size, in octets.";
          }
          leaf stream-blocked-due-to-oversize-frame-enabled {
            type boolean;
```

```
                default "false";
                description
                  "A value of TRUE indicates that the
                  StreamBlockedDueToOversizeFrame function is enabled; a
                  value of FALSE indicates that the
                  StreamBlockedDueToOversizeFrame function is disabled. The
                  default value of StreamBlockedDueToOversizeFrameEnable is
                  FALSE.";
                reference
                  "8.6.5.1 of IEEE Std 802.1Qcr
                  8.6.5.3.1 of IEEE Std 802.1Qcr";
              }
              leaf stream-blocked-due-to-oversize-frame {
                type boolean;
                default "false";
                config false;
                description
                  "If StreamBlockedDueToOversizeFrameEnable is TRUE, a
                  value of TRUE in StreamBlockedDueToOversizeFrame
                  indicates that all frames are to be dropped (i.e., the
                  behavior is identical as if the maximum SDU size would be
                  set to 0 octets). If StreamBlockedDueToOversizeFrame is
                  FALSE, it has no effect. The default value of
                  StreamBlockedDueToOversizeFrame is FALSE; if any frame is
                  discarded because it exceeds the maximum SDU size for the
                  stream, then StreamBlockedDueToOversizeFrame is set TRUE.";
                reference
                  "8.6.5.1 of IEEE Std 802.1Qcr
                  8.6.5.3.1 of IEEE Std 802.1Qcr";
              }
            }
          }
        }
        leaf max-stream-filter-instances {
          type uint32;
          config false;
          description
            "The maximum number of Stream Filter instances supported by this
            Bridge component.";
          reference
            "12.31.1.1 of IEEE Std 802.1Qcr
            8.6.5.1 of IEEE Std 802.1Qcr";
        }
      }
      container stream-gates {
        description
          "This container encapsulates all nodes related to Stream Gates.";
        list stream-gate-instance-table {
          key "stream-gate-instance-id";
          description
            "There is one Stream Gate Instance Table per Bridge component.
            Each table row contains a set of parameters that defines a single
            Stream Gate (8.6.5.1.2), as detailed in Table 12-32. Tables can
            be created or removed dynamically in implementations that support
            dynamic configuration of Bridge components. Rows in the table can
            be created or removed dynamically in implementations that support
            dynamic configuration of stream gates.";
          reference
```

```
            "12.31.3 of IEEE Std 802.1Qcr";
          leaf stream-gate-instance-id {
            type uint32;
            description
              "An integer table index that allows the stream gate to be
              referenced from Stream Filter Instance Table entries.";
            reference
              "12.31.2.4 of IEEE Std 802.1Qcr
              8.6.5.1 of IEEE Std 802.1Qcr
              8.6.5.2 of IEEE Std 802.1Qcr";
          }
          leaf gate-enable {
            type boolean;
            default "false";
            description
              "A Boolean variable that indicates whether the operation of the
              state machines is enabled (TRUE) or disabled (FALSE). This
              variable is set by management. The default value of this
              variable is FALSE.";
            reference
              "8.6.9.4.14 of IEEE Std 802.1Q-2018";
          }
          leaf admin-gate-states {
            type gate-state-value-type;
            default "open";
            description
              "The administrative state associated with this gate, as set by
              the management.";
            reference
              "12.31.3.2.1 of IEEE Std 802.1Qcr
              8.6.10.4 of IEEE Std 802.1Qcr";
          }
          leaf admin-ipv {
            type ipv-type;
            default "wildcard";
            description
              "The administrative internal priority value specification.";
            reference
              "12.31.3.3 of IEEE Std 802.1Qcr
              8.6.10.6 of IEEE Std 802.1Qcr
              8.6.5.2 of IEEE Std 802.1Qcr";
          }
        }
        leaf max-stream-gate-instances {
          type uint32;
          config false;
          description
            "The maximum number of Stream Gate instances supported by this
            Bridge component.";
          reference
            "12.31.1.2 of IEEE Std 802.1Qcr
            8.6.5.2 of IEEE Std 802.1Qcr";
        }
      }
    }
  }
}
```

<<Editor's Note: The discontinuous numbering of the previous and subsequent clause is intentional. A clause for PSFP appears between both (P802.1Qcw).>>

### 48.7.8 Definitions for the ieee802-dot1q-ats YANG module

```
module ieee802-dot1q-ats {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-ats;
  prefix ats;
  import ietf-yang-types {
    prefix yang;
  }
  import ietf-interfaces {
    prefix if;
  }
  import ieee802-dot1q-types {
    prefix dot1qtypes;
  }
  import ieee802-dot1q-bridge {
    prefix dot1q;
  }
  import ieee802-dot1q-stream-filters-gates {
    prefix sfsg;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: https://1.ieee802.org/
    WG-EMail: stds-802-1@ieee.org
    Contact: IEEE 802.1 Working Group Chair
    Postal: C/O IEEE 802.1 Working Group
            IEEE Standards Association
            445 Hoes Lane
            P.O. Box 1331
            Piscataway
            NJ 08855-1331
            USA

    E-mail: STDS-802-1-L@LISTSERV.IEEE.ORG";
  description
    "This module provides management of 802.1Q bridge components that
    support Asynchronous Traffic Shaping (ATS). NOTICE: This YANG module is
    part of an unapproved IEEE Standards Draft and is subject to change.";
  revision 2019-04-13 {
    description
      "Initial revision from IEEE P802.1Qcr.";
    reference
      "IEEE Std 802.1Qcr";
  }
  typedef scheduler-ref-type {
    type leafref {
      path
        '/dot1q:bridges'+
        '/dot1q:bridge'+
        '/dot1q:component'+
        '/ats:schedulers'+
        '/ats:scheduler-instance-table'+
        '/ats:scheduler-instance-id';
    }
    description
      "This type is used to refer to an ATS scheduler instance.";
```

```
  }
  typedef scheduler-group-ref-type {
    type leafref {
      path
        '/dot1q:bridges'+
        '/dot1q:bridge'+
        '/dot1q:component'+
        '/ats:scheduler-groups'+
        '/ats:scheduler-group-instance-table'+
        '/ats:scheduler-group-instance-id';
    }
    description
      "This type is used to refer to an ATS scheduler group instance.";
  }
  augment
    "/dot1q:bridges"+
    "/dot1q:bridge"+
    "/dot1q:component"+
    "/sfsg:stream-filters"+
    "/sfsg:stream-filter-instance-table"+
    "/sfsg:filter-specification-list"+
    "/sfsg:filter-specification" {
    description
      "Augments the Bridge component Stream Filter specification type by a
      ATS scheduler filter specification type.";
    case scheduler-ref {
      leaf scheduler-ref {
        type ats:scheduler-ref-type;
        mandatory true;
        description
          "A reference to the ATS scheduler associated with this filter.";
      }
    }
  }
  augment "/if:interfaces/if:interface/dot1q:bridge-port" {
    description
      "Augments Bridge Ports by ATS Per-Port Parameters";
    container ats-port-parameters {
      description
        "This container comprises all ATS Per-Port Parameters.";
      leaf discarded-frames-count {
        type yang:counter64;
        config false;
        description
          "A counter of frames discarded by ATS scheduler instances
          associated with the Bridge Port";
        reference
          "12.31.7.3 of IEEE Std 802.1Qcr";
      }
    }
  }
  augment "/dot1q:bridges/dot1q:bridge/dot1q:component" {
    description
      "Augments the Bridge component by
       a) ATS Schedulers
       b) ATS Scheduler Groups";
    container schedulers {
      description
        "This container comprises all ATS scheduler instance related nodes.";
```

```
    list scheduler-instance-table {
      key "scheduler-instance-id";
      description
        "Each table row in the Scheduler Instance Table comprises a set
        of parameters that defines a single ATS scheduler instance, as
        detailed in 8.6.5.2.3.";
      reference
        "12.31.5 of IEEE Std 802.1Qcr";
      leaf scheduler-instance-id {
        type uint32;
        mandatory true;
        description
          "An integer table index that allows the scheduler instance to
          be referenced from Stream Filter Instance Table entries.";
        reference
          "12.31.5.1 of IEEE Std 802.1Qcr
          8.6.5.3.3 of IEEE Std 802.1Qcr";
      }
      leaf committed-information-rate {
        type uint64;
        mandatory true;
        description
          "The committed information rate parameter of the scheduler
          instance, in bits per second.";
        reference
          "12.31.5.3 of IEEE Std 802.1Qcr
          8.6.5.3.3 of IEEE Std 802.1Qcr";
      }
      leaf committed-burst-size {
        type uint32;
        mandatory true;
        description
          "The committed burst size parameter of the scheduler instance,
          in bits.";
        reference
          "12.31.5.2 of IEEE Std 802.1Qcr
          8.6.5.3.3 of IEEE Std 802.1Qcr";
      }
      leaf scheduler-group-ref {
        type ats:scheduler-group-ref-type;
        mandatory true;
        description
          "The SchedulerGroupInstanceID parameter identifies the
          scheduler group (12.32.5) that is associated with the scheduler
          instance. Multiple scheduler instances can be associated to one
          scheduler group, as detailed in 8.6.5.2.3.";
        reference
          "12.31.6 of IEEE Std 802.1Qcr";
      }
    }
    leaf max-scheduler-instances {
      type uint32;
      config false;
      description
        "The maximum number of scheduler instances supported by this
        Bridge component.";
      reference
        "12.31.1.5 of IEEE Std 802.1Qcr
        8.6.5.3.3 of IEEE Std 802.1Qcr";
```

```
    }
  }
  container scheduler-groups {
    description
      "This container comprises all ATS scheduler group related nodes.";
    list scheduler-group-instance-table {
      key "scheduler-group-instance-id";
      description
        "Each table row in the Scheduler Group Instance Table comprises a
        set of parameters that defines a single ATS scheduler group
        instance (8.6.5.3.3).";
      reference
        "12.31.6 of IEEE Std 802.1Qcr
        8.6.5.3.3 of IEEE Std 802.1Qcr";
      leaf scheduler-group-instance-id {
        type uint32;
        description
          "An integer table index that allows the scheduler group
          instance to be referenced from Scheduler Instance Table
          entries.";
        reference
          "12.31.6.1 of IEEE Std 802.1Qcr
          8.6.5.3.3 of IEEE Std 802.1Qcr";
      }
      leaf max-residence-time {
        type uint32;
        mandatory true;
        description
          "The maximum residence time parameter of the scheduler group,
          in nanoseconds.";
        reference
          "8.6.11.2.13 of IEEE Std 802.1Qcr
          8.6.5.3.3 of IEEE Std 802.1Qcr";
      }
    }
    leaf max-scheduler-group-instances {
      type uint32;
      config false;
      description
        "The maximum number of scheduler group instances supported by
        this Bridge component.";
      reference
        "12.31.1.6 of IEEE Std 802.1Qcr
        8.6.5.3.3 of IEEE Std 802.1Qcr";
    }
    container scheduler-timing-characteristics {
      description
        "This container comprises all ATS scheduler timing
        characteristics related nodes.";
      list scheduler-timing-characteristics-table {
        key "reception-port transmission-port";
        config false;
        description
          "Each row in this table comprises the timing characteristics of
          a reception Port transmission Port pair, as detailed in Table
          12-36.";
        reference
          "12.31.8 of IEEE Std 802.1Qcr
          8.6.11 of IEEE Std 802.1Qcr";
```

```
        leaf reception-port {
          type dot1qtypes:port-number-type;
          config false;
          mandatory true;
          description
            "A reference to the associated reception Port.";
          reference
            "12.31.8.1 of IEEE Std 802.1Qcr";
        }
        leaf transmission-port {
          type dot1qtypes:port-number-type;
          config false;
          mandatory true;
          description
            "A reference to the associated transmission Port.";
          reference
            "12.31.8.2 of IEEE Std 802.1Qcr";
        }
        leaf clock-offset-variation-max {
          type uint32;
          config false;
          mandatory true;
          description
            "The maximum clock offset variation associated with the
            reception Port transmission Port pair, in nanoseconds.";
          reference
            "12.31.8.3 of IEEE Std 802.1Qcr";
        }
        leaf clock-rate-deviation-max {
          type uint32;
          config false;
          mandatory true;
          description
            "The maximum clock rate deviation associated with the
            reception Port transmission Port pair, in ppm.";
          reference
            "12.31.8.4 of IEEE Std 802.1Qcr";
        }
        leaf arrival-recognition-delay-max {
          type uint32;
          config false;
          mandatory true;
          description
            "The maximum arrival time recognition delay associated with
            the reception Port transmission Port pair, in nanoseconds.";
          reference
            "12.31.8.5 of IEEE Std 802.1Qcr";
        }
        leaf processing-delay-min {
          type uint32;
          config false;
          mandatory true;
          description
            "The minimum processing delay associated with the reception
            Port transmission Port pair, in nanoseconds.";
          reference
            "12.31.8.6 of IEEE Std 802.1Qcr";
        }
        leaf processing-delay-max {
```

```
            type uint32;
            config false;
            mandatory true;
            description
              "The maximum processing delay associated with the reception
              Port transmission Port pair, in nanoseconds.";
            reference
              "12.31.8.7 of IEEE Std 802.1Qcr";
          }
        }
      }
    }
  }
}
```