

Contents

1.	Overview.....	15
1.3	Introduction.....	15
12.	Bridge management	16
12.29	Managed objects for scheduled traffic.....	16
12.29.1	The Gate Parameter Table	16
12.31	Managed objects for per-stream classification and metering	17
12.31.1	The Stream Parameter Table	17
12.31.3	The Stream Gate Instance Table	17
17.	Management Information Base (MIB)	18
17.2	Structure of the MIB	18
17.2.22	Structure of the IEEE8021-ST-MIB	18
17.2.24	Structure of the IEEE8021-PSFP-MIB	18
17.7	MIB modules	19
17.7.14	Definitions for the IEEE8021-SRP-MIB module	19
17.7.22	Definitions for the IEEE8021-ST-MIB module	38
17.7.24	Definitions for the IEEE8021-PSFP-MIB module	49
35.	Stream Reservation Protocol (SRP).....	69
35.2	Definition of the MSRP application	69
35.2.2	Definition of MRP elements	69
46.	Time-Sensitive Networking (TSN) configuration	70
46.2	User/network configuration information	70
46.2.3	Talker	70
48.	YANG Data Models	71
48.2	IEEE 802.1Q YANG models.....	71
48.2.1	VLAN Bridge components model	71
48.2.2	Two-Port MAC Relay (TPMR) model	73
48.2.4	Provider Bridge model	74
48.2.5	CFM Model	77
48.2.7	Asynchronous Traffic Shaping (ATS) model	79
48.2.8	Congestion Isolation (CI) model	80
48.2.9	Scheduled Traffic model	80
48.2.10	Frame Preemption model	81
48.2.11	Per-Stream Filtering and Policing model	82
48.3	Structure of the YANG models	84
48.3.9	Scheduled Traffic model	84
48.3.10	Frame Preemption model	85
48.3.11	Per-Stream Filtering and Policing model	85
48.4	Security considerations	86
48.4.9	Security considerations of the Scheduled Traffic model	86
48.4.10	Security considerations of the Frame Preemption model	86
48.4.11	Security considerations of the Per-Stream Filtering and Policing model	86

48.5	YANG schema tree definitions	87
48.5.4	Schema for the ieee802-dot1q-bridge YANG module	87
48.5.17	Schema for the ieee802-dot1q-sched YANG module	94 91
48.5.18	Schema for the ieee802-dot1q-sched-bridge YANG module	94 91
48.5.19	Schema for the ieee802-dot1q-preemption YANG module	95 91
48.5.20	Schema for the ieee802-dot1q-preemption-bridge YANG module	95 92
48.5.21	Schema for the ieee802-dot1q-psfp YANG module	95 92
48.5.22	Schema for the ieee802-dot1q-psfp-bridge YANG module	95 92
48.6	YANG modules	97 94
48.6.1	The ieee802-types YANG module	97 94
48.6.2	The ieee802-dot1q-types YANG module	102 99
48.6.4	The ieee802-dot1q-bridge YANG module	118 115
48.6.6	The ieee802-dot1q-pb YANG module	144 141
48.6.17	The ieee802-dot1q-sched YANG module	148 145
48.6.18	The ieee802-dot1q-sched-bridge YANG module	154 151
48.6.19	The ieee802-dot1q-preemption YANG module	155 152
48.6.20	The ieee802-dot1q-preemption-bridge YANG module	159 156
48.6.21	The ieee802-dot1q-psfp YANG module	160 157
48.6.22	The ieee802-dot1q-psfp-bridge YANG module	170 167
Annex A (normative)	PICS proforma—Bridge implementations	180 168
A.47	YANG	180 168
Annex B (normative)	PICS proforma—End station implementations	181 169
B.15	Scheduled traffic	181 169
B.16	Frame Preemption.....	181 169
B.17	Per-Stream Filtering and Policing.....	181 169

48.5 YANG schema tree definitions

48.5.4 Schema for the ieee802-dot1q-bridge YANG module

Delete the existing text in 48.5.4.

Insert new text in 48.5.4 as follows:

```
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 bridge-name? -> /bridges/bridge/name
  | +--rw component-name? ->
  /bridges/bridge[dot1q:name=current()/../bridge-name]/component/name
  +--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 pcsp-selection? dot1qtypes:pcsp-selection-type
  +--rw pcsp-decoding-table
  | +--rw pcsp-decoding-map* [pcsp]
  | | +--rw pcsp pcsp-selection-type
  | | +--rw priority-map* [priority-code-point]
  | | | +--rw priority-code-point priority-type
  | | | +--rw priority? priority-type
  | | | +--rw drop-eligible? boolean
  +--rw pcsp-encoding-table
  | +--rw pcsp-encoding-map* [pcsp]
  | | +--rw pcsp pcsp-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
| x--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 traffic-class-table
| | +--rw number-of-traffic-classes?    uint8
| | +--rw priority0?                   traffic-class-type
| | +--rw priority1?                   traffic-class-type
| | +--rw priority2?                   traffic-class-type
| | +--rw priority3?                   traffic-class-type
| | +--rw priority4?                   traffic-class-type
| | +--rw priority5?                   traffic-class-type
| | +--rw priority6?                   traffic-class-type
| | +--rw priority7?                   traffic-class-type
+--rw transmission-selection-algorithm-table
| +--rw transmission-selection-algorithm-map* [traffic-class]
| | +--rw traffic-class          traffic-class-type
| | +--rw transmission-selection-algorithm?    identityref
+--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-capabilities?     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

module: ieee802-dot1q-stream-filters-gates-bridge
--augment /dot1q:bridges/dot1q:bridge/dot1q:component:
-- +--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-spec-type
+--ro max-stream-gate-instances? uint32
+--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
+--rw priority-spec-type
+--rw max-sdu-size uint32
+--rw stream-blocked-due-to-oversize-frame-enabled? boolean
+--rw stream-blocked-due-to-oversize-frame? boolean
+--rw stream-gate-ref leafref
+--ro max-stream-filter-instances? uint32
module: ieee802-dot1q-ats
+--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-spec-type
+--ro max-stream-gate-instances? uint32
+--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
+--rw priority-spec-type
+--rw max-sdu-size uint32
+--rw stream-blocked-due-to-oversize-frame-enabled? boolean
+--rw stream-blocked-due-to-oversize-frame? boolean
+--rw stream-gate-ref leafref
+--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 leafref
+--ro max-scheduler-instances? uint32
+--rw scheduler-groups
+--rw scheduler-group-instance-table*
+--rw scheduler-group-instance-id [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*
+--rw {reception-port-transmission-port}
+--ro reception-port
+--rw dot1qtypes:port-number-type
+--ro transmission-port
+--rw 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
+--rw scheduler
+--rw scheduler-ref? leafref
+--rw scheduler-enable? boolean
+--ro max-stream-filter-instances? uint32
module: ieee802-dot1q-ats-bridge

```

```

--augment /if:interfaces/if:interface/dot1q:bridge-port:
--  +--rw discarded-frames-count? yang:counter64
--augment /dot1q:bridges/dot1q:bridge/dot1q:component:
--  +--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-spec-type
--    | | +--ro max-stream-gate-instances? uint32
--  +--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
--    | | | priority-spec-type
--    | | +--rw max-sdu-size uint32
--    | | +--rw stream-blocked-due-to-oversize-frame-enabled? boolean
--    | | +--rw stream-blocked-due-to-oversize-frame? boolean
--    | | +--rw stream-gate-ref leafref
--    | +--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 leafref
--    | | | +--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
--    | | +--rw scheduler
--    | | | +--rw scheduler-ref? leafref
--    | | | +--rw scheduler-enable? boolean
--    | +--ro max-stream-filter-instances? uint32

module: ieee802-dot1q-congestion-isolation
--  +--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-spec-type
--    | | +--ro max-stream-gate-instances? uint32
--  +--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
--    | | | | +--:(null-handle) {congestion-isolation}?
--    | | | | +--rw null-handle? empty

```



```

| +--rw priority-spec
| | priority-spec-type
| +--rw max-sdu-size uint32
| +--rw stream-blocked-due-to-oversize-frame-enabled? boolean
| +--rw stream-blocked-due-to-oversize-frame? boolean
| +--rw stream-gate-ref leafref
+--ro max-stream-filter-instances? uint32

module: ieee802-dot1q-congestion-isolation-bridge
augment /dot1q:bridges/dot1q:bridge/dot1q:component:
+--rw ci-master-enable? boolean {congestion-isolation-bridge}?
+--ro ci-cim-tx-priority? dot1q-types:priority-type
| {congestion-isolation-bridge}?
+--ro ci-max-flow-life? uint32 {congestion-isolation-bridge}?
+--rw ci-peers {congestion-isolation-bridge}?
| +--rw ci-peer-table* [reception-port]
| | +--rw reception-port dot1q-types:port-number-type
| | | +--rw cim-type? enumeration
| | | +--rw peer-mac-address? ieee:mac-address
| | | +--rw peer-ipv4-address? inet:ipv4-address
| | | +--rw peer-ipv6-address? inet:ipv6-address
| | | +--rw peer-udp-port? inet:port-number
| | | +--rw peer-cim-encap-len? uint16
| | +--ro max-ci-peer-entries? uint32
+--rw ci-streams {congestion-isolation-bridge}?
| +--ro ci-stream-table* [stream-handle-id]
| | +--ro stream-handle-id uint32
| | | +--ro cim-count? uint16
| | | +--ro create-time? yang:timeticks
| | | +--ro create-mask? bits
| | +--ro queue-key? uint16
| | +--ro dest-mac-address? ieee:mac-address
| | +--ro source-mac-address? ieee:mac-address
| | +--ro vid? dot1q-types:vlan-index-type
| | +--ro msdu? yang:hex-string
+--ro max-ci-stream-entries? uint32 {congestion-isolation-bridge}?
+--rw stream-gates {congestion-isolation-bridge}?
| +--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-spec-type
| +--ro max-stream-gate-instances? uint32
+--rw stream-filters {congestion-isolation-bridge}?
+--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
| | | +--:(null-handle) {congestion-isolation-bridge}?
| | | +--rw null-handle? empty
| +--rw priority-spec
| | priority-spec-type
| +--rw max-sdu-size uint32
| +--rw stream-blocked-due-to-oversize-frame-enabled? boolean
| +--rw stream-blocked-due-to-oversize-frame? boolean
| +--rw stream-gate-ref leafref
+--ro max-stream-filter-instances? uint32
augment /if:interfaces/if:interface/dot1q:bridge-port:
+--ro cip-mac-address? ieee:mac-address
| {congestion-isolation-bridge}?
+--ro cip-ipv4-address? inet:ipv4-address
| {congestion-isolation-bridge}?
+--ro cip-ipv6-address? inet:ipv6-address
| {congestion-isolation-bridge}?
+--ro cip-cim-port? inet:port-number
| {congestion-isolation-bridge}?
+--rw queue-map* [priority] {congestion-isolation-bridge}?

```

```

| +--rw priority                               dot1q-types:priority-type
| +--rw abs-traffic-class-plus-one?
|         abs-traffic-class-plus-one-type
+--rw min-header-octets? uint16 {congestion-isolation-bridge}?
+--rw max-cim-tx?       uint16 {congestion-isolation-bridge}?

```

Insert new 48.5.17, 48.5.18, 48.5.19, 48.5.20, 48.5.21, and 48.5.22 at the end of 48.5, as follows:

48.5.17 Schema for the ieee802-dot1q-sched YANG module

This YANG module does not have a YANG schema tree.

48.5.18 Schema for the ieee802-dot1q-sched-bridge YANG module

module: ieee802-dot1q-sched-bridge

```

augment /if:interfaces/if:interface/dot1q:bridge-port:
  +--rw gate-parameter-table
    +--rw queue-max-sdu-table* [traffic-class]
      | +--rw traffic-class          dot1q-types:traffic-class-type
      | +--rw queue-max-sdu?        uint32
      | +--ro transmission-overflow? yang:counter64
    +--rw gate-enabled?            boolean
    +--rw admin-gate-states?        uint8
    +--ro oper-gate-states?         uint8
    +--rw admin-control-list
      | +--rw gate-control-entry* [index]
      |   +--rw index              uint32
      |   +--rw operation-name      identityref
      |   +--rw time-interval-value? uint32
      |   +--rw gate-states-value   uint8
    +--ro oper-control-list
      | +--ro gate-control-entry* [index]
      |   +--ro index              uint32
      |   +--ro operation-name      identityref
      |   +--ro time-interval-value? uint32
      |   +--ro gate-states-value   uint8
    +--rw admin-cycle-time
      | +--rw numerator?           uint32
      | +--rw denominator?         uint32
    +--ro oper-cycle-time
      | +--ro numerator?           uint32
      | +--ro denominator?         uint32
    +--rw admin-cycle-time-extension? uint32
    +--ro oper-cycle-time-extension?  uint32
    +--rw admin-base-time
      | +--rw seconds?             uint64
      | +--rw nanoseconds?         uint32
    +--ro oper-base-time
      | +--ro seconds?             uint64
      | +--ro nanoseconds?         uint32
    +--rw config-change?            boolean
    +--ro config-change-time
      | +--ro seconds?             uint64
      | +--ro nanoseconds?         uint32
    +--ro tick-granularity?         uint32
    +--ro current-time
      | +--ro seconds?             uint64
      | +--ro nanoseconds?         uint32
    +--ro config-pending?           boolean
    +--ro config-change-error?      yang:counter64
    +--rw supported-list-max?       uint32
    +--rw supported-cycle-max
      | +--rw numerator?           uint32
      | +--rw denominator?         uint32
    +--rw supported-interval-max?   uint32

```

48.5.19 Schema for the ieee802-dot1q-preemption YANG module

This YANG module does not have a YANG schema tree.

48.5.20 Schema for the ieee802-dot1q-preemption-bridge YANG module

module: ieee802-dot1q-preemption-bridge

```
augment /if:interfaces/if:interface/dot1q:bridge-port:
  +--rw frame-preemption-parameters {frame-preemption}?
    +--rw frame-preemption-status-table
      | +--rw priority0?   frame-preemption-status-enum
      | +--rw priority1?   frame-preemption-status-enum
      | +--rw priority2?   frame-preemption-status-enum
      | +--rw priority3?   frame-preemption-status-enum
      | +--rw priority4?   frame-preemption-status-enum
      | +--rw priority5?   frame-preemption-status-enum
      | +--rw priority6?   frame-preemption-status-enum
      | +--rw priority7?   frame-preemption-status-enum
      +--ro hold-advance?   uint32
      +--ro release-advance? uint32
      +--ro preemption-active? boolean
      +--ro hold-request?   enumeration
```

48.5.21 Schema for the ieee802-dot1q-psfp YANG module

This YANG module does not have a YANG schema tree.

48.5.22 Schema for the ieee802-dot1q-psfp-bridge YANG module

module: ieee802-dot1q-psfp-bridge

```
augment /dot1q:bridges/dot1q:bridge/dot1q:component:
  +--rw flow-meters
    | +--rw flow-meter-instance-table* [flow-meter-instance-id]
    | | +--rw flow-meter-instance-id   uint32
    | | +--rw committed-information-rate uint64
    | | +--rw committed-burst-size     uint32
    | | +--rw excess-information-rate   uint64
    | | +--rw excess-burst-size         uint32
    | | +--rw coupling-flag             enumeration
    | | +--rw color-mode                 enumeration
    | | +--rw drop-on-yellow             boolean
    | | +--rw mark-all-frames-red-enable? boolean
    | | +--rw mark-all-frames-red?     boolean
    | +--rw max-flow-meter-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-spec-type
    | | +--ro oper-gate-state?           sfsg:gate-state-value-type
    | | +--rw oper-ipv?                  sfsg:ipv-spec-type
    | | +--rw admin-control-list
    | | | +--rw gate-control-entry* [index]
    | | | | +--rw index                 uint32
    | | | | +--rw operation-name         identityref
    | | | | +--rw time-interval-value?   uint32
    | | | | +--rw gate-state-value       sfsg:gate-state-value-type
    | | | | +--rw ipv-spec               sfsg:ipv-spec-type
    | | | | +--rw interval-octet-max?    uint32
    | | +--ro oper-control-list
    | | | +--ro gate-control-entry* [index]
    | | | | +--ro index                 uint32
    | | | | +--ro operation-name         identityref
    | | | | +--ro time-interval-value?   uint32
    | | | | +--ro gate-state-value       sfsg:gate-state-value-type
```

IEEE Std 802.1Qcw™-2023
IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks
Amendment 36: YANG Data Models for Scheduled Traffic, Frame Preemption, and Per-Stream Filtering and Policing

```

| | | +--ro ipv-spec sfsg:ipv-spec-type
| | | +--ro interval-octet-max? uint32
| | +--rw admin-cycle-time
| | | +--rw numerator? uint32
| | | +--rw denominator? uint32
| | +--ro oper-cycle-time
| | | +--ro numerator? uint32
| | | +--ro denominator? uint32
| | +--rw admin-cycle-time-extension? uint32
| | +--ro oper-cycle-time-extension? uint32
| | +--rw admin-base-time
| | | +--rw seconds? uint64
| | | +--rw nanoseconds? uint32
| | +--ro oper-base-time
| | | +--ro seconds? uint64
| | | +--ro nanoseconds? uint32
| | +--rw config-change? boolean
| | +--ro config-change-time
| | | +--ro seconds? uint64
| | | +--ro nanoseconds? uint32
| | +--ro tick-granularity? uint32
| | +--ro current-time
| | | +--ro seconds? uint64
| | | +--ro nanoseconds? uint32
| | +--ro config-pending? boolean
| | +--ro config-change-error? yang:counter64
| | +--rw gate-closed-due-to-invalid-rx-enable? boolean
| | +--rw gate-closed-due-to-invalid-rx? boolean
| | +--rw gate-closed-due-octets-exceeded-enable? boolean
| | +--rw gate-closed-due-octets-exceeded? boolean
| +--ro max-stream-gate-instances? uint32
| +--rw supported-list-max? uint32
| +--rw supported-cycle-max
| | +--rw numerator? uint32
| | +--rw denominator? uint32
| +--rw supported-interval-max? uint32
+--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 priority-spec-type
    | +--rw max-sdu-size uint32
    | +--rw stream-blocked-due-to-oversize-frame-enabled? boolean
    | +--rw stream-blocked-due-to-oversize-frame? boolean
    | +--rw stream-gate-ref ->
  ..../stream-gates/stream-gate-instance-table/stream-gate-instance-id
    | +--ro matching-frames-count? yang:counter64
    | +--ro passing-frames-count? yang:counter64
    | +--ro not-passing-frames-count? yang:counter64
    | +--ro red-frames-count? yang:counter64
    | +--ro passing-sdu-count? yang:counter64
    | +--ro not-passing-sdu-count? yang:counter64
    | +--rw flow-meter-ref? ->
  ..../flow-meters/flow-meter-instance-table/flow-meter-instance-id
    | +--rw flow-meter-enable? boolean
    +--ro max-stream-filter-instances? uint32

```

48.6 YANG modules^{9 10 11}

48.6.1 The ieee802-types YANG module

Delete the YANG module in 48.6.1.

Insert the following YANG module in 48.6.1:

```
module ieee802-types {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-types;
  prefix ieee;
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: stds-802-1-chairs@ieee.org";
  description
    "This module contains a collection of generally useful derived data
    types for IEEE YANG models.

    Copyright (C) IEEE (2023). All rights reserved.

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-04-10-19-22 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE Std 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023-.
      IEEE Std 802 Overview and Architecture:
      IEEE Std 802-2014.
      IEEE 802.1AS Timing and Synchronization for Time-Sensitive Applications:
      IEEE Std 802.1AS-2020.";
  }
  revision 2022-10-29 {
    description
      "Published as part of IEEE Std 802.1Q-2022.";
    reference
      "IEEE Std 802.1Q-2022, Bridges and Bridged Networks.";
  }
  revision 2022-03-16 {
    description
      "Published as part of IEEE Std 802.1ABcu.";
    reference
      "IEEE Std 802.1AB-2016";
  }
  revision 2020-06-04 {
    description
      "Published as part of IEEE Std 802.1Qcx-2020. Second version.";
```

⁹ Copyright release for YANG: Users of this standard may freely reproduce the YANG modules contained in this standard so that they can be used for their intended purpose.

¹⁰ An ASCII version of each YANG module is attached to the PDF of this standard and can also be obtained from the IEEE 802.1 Website at <https://1.ieee802.org/yang-modules/>.

¹¹ References in this standard's YANG module definitions are not clickable, as each module has been incorporated unchanged after development and verification using YANG tools.

```
reference
  "IEEE Std 802.1Qcx-2020, Bridges and Bridged Networks - YANG Data
  Model for Connectivity Fault Management.";
}
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 hexadecimal
    representation uses uppercase characters.";
  reference
    "3.1, 8.1 of IEEE Std 802";
}
typedef chassis-id-subtype-type {
  type enumeration {
    enum chassis-component {
      value 1;
      description
        "Represents a chassis identifier based on the value of the
        entPhysicalAlias object (defined in IETF RFC 2737) for a chassis
        component (i.e., an entPhysicalClass value of chassis(3)).";
    }
    enum interface-alias {
      value 2;
      description
        "Represents a chassis identifier based on the value of the ifAlias
        object (defined in IETF RFC 2863) for an interface on the
        containing chassis.";
    }
    enum port-component {
      value 3;
      description
        "Represents a chassis identifier based on the value of the
        entPhysicalAlias object (defined in IETF RFC 2737) for a port or
        backplane component (i.e., entPhysicalClass value of port(10) or
        backplane(4)), within the containing chassis.";
    }
    enum mac-address {
      value 4;
      description
        "Represents a chassis identifier based on the value of a unicast
        source address (encoded in network byte order and IEEE 802.3
        bit significance), of a port on the containing chassis as
        defined in IEEE Std 802.";
    }
    enum network-address {
      value 5;
      description
        "Represents a chassis identifier based on a network address,
        associated with a particular chassis. The encoded address is
        actually composed of two fields. The first field is a single
        octet, representing the IANA AddressFamilyNumbers value for the
        specific address type, and the second field is the network address
        value.";
    }
    enum interface-name {
      value 6;
      description
        "Represents a chassis identifier based on the value of the ifName
        object (defined in IETF RFC 2863) for an interface on the
        containing chassis.";
    }
    enum local {
      value 7;
    }
  }
}
```

```
        description
            "Represents a chassis identifier based on a locally defined value.";
    }
}
description
    "The source of a chassis identifier.";
reference
    "IEEE Std 802
    IETF RFC 2737
    IETF RFC 2863";
}
typedef chassis-id-type {
    type string {
        length "1..255";
    }
    description
        "The format of a chassis identifier string. Objects of this type are
        always used with an associated chassis-id-subtype object, which
        identifies the format of the particular chassis-id object
        instance.

        If the associated chassis-id-subtype object has a value of
        chassis-component, then the octet string identifies a particular
        instance of the entPhysicalAlias object (defined in IETF RFC 2737) for
        a chassis component (i.e., an entPhysicalClass value of chassis(3)).

        If the associated chassis-id-subtype object has a value of
        interface-alias, then the octet string identifies a particular
        instance of the ifAlias object (defined in IETF RFC 2863) for an
        interface on the containing chassis. If the particular ifAlias object
        does not contain any values, another chassis identifier type should be
        used.

        If the associated chassis-id-subtype object has a value of
        port-component, then the octet string identifies a particular instance
        of the entPhysicalAlias object (defined in IETF RFC 2737) for a port
        or backplane component within the containing chassis.

        If the associated chassis-id-subtype object has a value of
        mac-address, then this string identifies a particular unicast source
        address (encoded in network byte order and IEEE 802.3 bit
        significance), of a port on the containing chassis as defined in
        IEEE Std 802.

        If the associated chassis-id-subtype object has a value of
        network-address, then this string identifies a particular network
        address, encoded in network byte order, associated with one or more
        ports on the containing chassis. The first octet contains the IANA
        Address Family Numbers enumeration value for the specific address
        type, and octets 2 through N contain the network address value in
        network byte order.

        If the associated chassis-id-subtype object has a value of
        interface-name, then the octet string identifies a particular instance
        of the ifName object (defined in IETF RFC 2863) for an interface on
        the containing chassis. If the particular ifName object does not
        contain any values, another chassis identifier type should be used.

        If the associated chassis-id-subtype object has a value of local,
        then this string identifies a locally assigned Chassis ID.";
    reference
        "IEEE Std 802
        IETF RFC 2737
        IETF RFC 2863";
}
typedef port-id-subtype-type {
    type enumeration {
        enum interface-alias {
            value 1;
            description
                "Represents a port identifier based on the ifAlias MIB object,
                defined in IETF RFC 2863.";
```

```

    }
    enum port-component {
      value 2;
      description
        "Represents a port identifier based on the value of
        entPhysicalAlias (defined in IETF RFC 2737) for a port component
        (i.e., entPhysicalClass value of port(10)), within the containing
        chassis.";
    }
    enum mac-address {
      value 3;
      description
        "Represents a port identifier based on a unicast source address
        (encoded in network byte order and IEEE 802.3 bit significance),
        which has been detected by the agent and associated with a
        particular port (IEEE Std 802).";
    }
    enum network-address {
      value 4;
      description
        "Represents a port identifier based on a network address, detected
        by the agent and associated with a particular port.";
    }
    enum interface-name {
      value 5;
      description
        "Represents a port identifier based on the ifName MIB object,
        defined in IETF RFC 2863.";
    }
    enum agent-circuit-id {
      value 6;
      description
        "Represents a port identifier based on the agent-local identifier
        of the circuit (defined in RFC 3046), detected by the agent and
        associated with a particular port.";
    }
    enum local {
      value 7;
      description
        "Represents a port identifier based on a value locally assigned.";
    }
  }
  description
    "The source of a particular type of port identifier.";
  reference
    "IEEE Std 802
    IETF RFC 2737
    IETF RFC 2863
    IETF RFC 3046";
}
typedef port-id-type {
  type string {
    length "1..255";
  }
}
description
  "The format of a port identifier string. Objects of this type are
  always used with an associated port-id-subtype object, which
  identifies the format of the particular port-id object instance.

  If the associated port-id-subtype object has a value of
  interface-alias, then the octet string identifies a particular
  instance of the ifAlias object (defined in IETF RFC 2863). If the
  particular ifAlias object does not contain any values, another port
  identifier type should be used.

  If the associated port-id-subtype object has a value of
  port-component, then the octet string identifies a particular instance
  of the entPhysicalAlias object (defined in IETF RFC 2737) for a port
  or backplane component.

  If the associated port-id-subtype object has a value of
  mac-address, then this string identifies a particular unicast source

```


address (encoded in network byte order and IEEE 802.3 bit significance) associated with the port (IEEE Std 802).

If the associated port-id-subtype object has a value of network-address, then this string identifies a network address associated with the port. The first octet contains the IANA AddressFamilyNumbers enumeration value for the specific address type, and octets 2 through N contain the networkAddress address value in network byte order.

If the associated port-id-subtype object has a value of interface-name, then the octet string identifies a particular instance of the ifName object (defined in IETF RFC 2863). If the particular ifName object does not contain any values, another port identifier type should be used.

If the associated port-id-subtype object has a value of agent-circuit-id, then this string identifies a agent-local identifier of the circuit (defined in RFC 3046).

If the associated port-id-subtype object has a value of local, then this string identifies a locally assigned port ID.";

```
reference
  "IEEE Std 802
  IETF RFC 2737
  IETF RFC 2863
  IETF RFC 3046";
}
grouping rational-grouping {
  description
    "Definition of a non-negative rational number.";
  leaf numerator {
    type uint32;
    description
      "Numerator of the rational number.";
  }
  leaf denominator {
    type uint32 {
      range "1..4294967295";
    }
    description
      "Denominator of the rational number.";
  }
}
grouping ptp-time-grouping {
  description
    "This grouping specifies a PTP timestamp, represented as a 48-bit
    unsigned integer number of seconds and a 32-bit unsigned integer
    number of nanoseconds.";
  reference
    "6.3.4.3.4 of IEEE Std 802.1AS";
  leaf seconds {
    type uint64;
    description
      "This is the integer portion of the timestamp in units of seconds.
      The upper 16 bits are always zero.";
  }
  leaf nanoseconds {
    type uint32;
    description
      "This is the fractional portion of the timestamp in units of
      nanoseconds. This value is always less than 10^9.";
  }
}
}
```

48.6.2 The ieee802-dot1q-types YANG module

Delete the YANG module in 48.6.2.

Insert the following YANG module in 48.6.2:

```
module ieee802-dot1q-types {
  yang-version "1.1";
  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://ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: stds-802-1-chairs@ieee.org";
  description
    "Common types used within dot1Q-bridge modules.

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-03-10-08-22 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE Std Std 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023.
      IEEE Std 802 Overview and Architecture:
      IEEE Std 802-2014.";
  }
  revision 2023-01-07-24-03 {
    description
      "Published as part of IEEE Std 802.1Qcz-2023.";
    reference
      "IEEE Std 802.1Qcz-2023, Bridges and Bridged Networks - Congestion
      Isolation.";
  }
  revision 2022-10-29 {
    description
      "Published as part of IEEE Std 802.1Q-2022.";
    reference
      "IEEE Std 802.1Q-2022, Bridges and Bridged Networks.";
  }
  revision 2020-06-04 {
    description
      "Published as part of IEEE Std 802.1Qcx-2020.";
    reference
      "IEEE Std 802.1Qcx-2020, Bridges and Bridged Networks - YANG Data
      Model for Connectivity Fault Management.";
  }
  revision 2018-03-07 {
    description
      "Published as part of IEEE Std 802.1Q-2018.";
    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 fromderived.";
  }
  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";
  }
  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";
  }
  identity transmission-selection-algorithm {
    description
      "Specify the transmission selection algorithms of IEEE Std 802.1Q
      Table 8-6";
  }
  identity strict-priority {
    base transmission-selection-algorithm;
    description
      "Indicates the strict priority transmission selection algorithm.";
    reference
      "Table 8-6 of IEEE Std 802.1Q";
  }
  identity credit-based-shaper {
    base transmission-selection-algorithm;
    description
      "Indicates the creditcredit-based shaper transmission selection algorithm.";
    reference
      "Table 8-6 of IEEE Std 802.1Q";
  }
  identity enhanced-transmission-selection {
    base transmission-selection-algorithm;
    description
      "Indicates the enhanced transmission selection algorithm.";
    reference
      "Table 8-6 of IEEE Std 802.1Q";
  }
  identity asynchronous-traffic-shaping {
    base transmission-selection-algorithm;
    description
      "Indicates the asynchronous transmission selection algorithm.";
    reference
      "Table 8-6 of IEEE Std 802.1Q";
  }
  identity vendor-specific {
    base transmission-selection-algorithm;
    description
      "Indicates a vendor specific transmission selection algorithm.";
    reference
      "Table 8-6 of IEEE Std 802.1Q";
  }
  identity type-of-operation {
    description
      "Represents the operation type (name).";
  }
  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..4095";
  }
  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 num-traffic-class-type {
  type uint8 {
    range "1..8";
  }
  description
    "The number of traffic classes supported or participating in a
    particular feature. There are between 1 and 8 supported traffic
    classes defined by IEEE Std 802.1Q.";
}
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 idsIDs, or non overlapping VLAN ranges, in ascending
    order, between 1 and 4094.

    This type is used to match an ordered list of VLAN idsIDs, or contiguous
    ranges of VLAN idsIDs. Valid VLAN idsIDs 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";
}
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";
}
typedef pcsp-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, 6.9.3 of IEEE Std 802.1Q";
}
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";
}
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";
}
typedef dot1q-tag-type {
```

```
type identityref {
  base dot1q-vlan-type;
}
description
  "Identifies a specific 802.1Q tag type";
reference
  "9.5 IEEE Std 802.1Q";
}
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
    "3.273 of IEEE Std 802.1Q";
}
grouping dot1q-tag-classifier-grouping {
  description
    "A grouping which represents an 802.1Q VLAN, matching both the
    EtherType and a single VLAN IdID.";
  leaf tag-type {
    type dot1q-tag-type;
    mandatory true;
    description
      "VLAN type";
  }
  leaf vlan-id {
    type vlanid;
    mandatory true;
    description
      "VLAN IdID";
  }
}
grouping dot1q-tag-or-any-classifier-grouping {
  description
    "A grouping which represents an 802.1Q VLAN, matching both the
    EtherType and a single VLAN IdID or 'any' to match on any VLAN IdID.";
  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 IdID match";
        }
      }
    }
    mandatory true;
    description
      "VLAN IdID or any";
  }
}
grouping dot1q-tag-ranges-classifier-grouping {
  description
    "A grouping which represents an 802.1Q VLAN that matches a range of
    VLAN IdIDs.";
  leaf tag-type {
    type dot1q-tag-type;
    mandatory true;
    description
      "VLAN type";
  }
}
```

```

    }
    leaf vlan-ids {
      type vid-range-type;
      mandatory true;
      description
        "VLAN IdsIDs";
    }
  }
  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 IdID, ordered list of ranges, or 'any' to
      match on any VLAN IdID.";
    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 IdsIDs 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";
    leaf priority0 {
      type priority-type;
      default "0";
      description
        "Priority 0";
      reference
        "12.6.2.3, 6.9.4 of IEEE Std 802.1Q";
    }
    leaf priority1 {
      type priority-type;
      default "1";
      description
        "Priority 1";
      reference
        "12.6.2.3, 6.9.4 of IEEE Std 802.1Q";
    }
    leaf priority2 {
      type priority-type;
      default "2";
      description
        "Priority 2";
      reference
        "12.6.2.3, 6.9.4 of IEEE Std 802.1Q";
    }
    leaf priority3 {
      type priority-type;
      default "3";
      description
        "Priority 3";
      reference
        "12.6.2.3, 6.9.4 of IEEE Std 802.1Q";
    }
  }

```

```

}
leaf priority4 {
  type priority-type;
  default "4";
  description
    "Priority 4";
  reference
    "12.6.2.3, 6.9.4 of IEEE Std 802.1Q";
}
leaf priority5 {
  type priority-type;
  default "5";
  description
    "Priority 5";
  reference
    "12.6.2.3, 6.9.4 of IEEE Std 802.1Q";
}
leaf priority6 {
  type priority-type;
  default "6";
  description
    "Priority 6";
  reference
    "12.6.2.3, 6.9.4 of IEEE Std 802.1Q";
}
leaf priority7 {
  type priority-type;
  default "7";
  description
    "Priority 7";
  reference
    "12.6.2.3, 6.9.4 of IEEE Std 802.1Q";
}
}
grouping pcg-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";
  list pcg-decoding-map {
    key "pcg";
    description
      "This map associates the priority code point field found in the VLAN
      to with a priority and drop eligible value based upon the priority code
      point selection type.";
    leaf pcg {
      type pcg-selection-type;
      description
        "The priority code point selection type.";
      reference
        "12.6.2.7, 6.9.3 of IEEE Std 802.1Q";
    }
    list priority-map {
      key "priority-code-point";
      description
        "This map associated associates a priority code point value to priority and with
        priority and drop eligible parameters.";
      leaf priority-code-point {
        type priority-type;
        description
          "Priority associated with the pcg.";
        reference
          "12.6.2.7, 6.9.3 of IEEE Std 802.1Q";
      }
      leaf priority {
        type priority-type;
        description
          "Priority associated with the pcg.";
        reference
          "12.6.2.7, 6.9.3 of IEEE Std 802.1Q";
      }
    }
  }
}

```



```

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

```

```

    reference
      "12.6.2.17, 6.13.1 of IEEE Std 802.1Q";
  }
  leaf priority1 {
    type priority-type;
    default "1";
    description
      "Service access priority value for priority 1";
    reference
      "12.6.2.17, 6.13.1 of IEEE Std 802.1Q";
  }
  leaf priority2 {
    type priority-type;
    default "2";
    description
      "Service access priority value for priority 2";
    reference
      "12.6.2.17, 6.13.1 of IEEE Std 802.1Q";
  }
  leaf priority3 {
    type priority-type;
    default "3";
    description
      "Service access priority value for priority 3";
    reference
      "12.6.2.17, 6.13.1 of IEEE Std 802.1Q";
  }
  leaf priority4 {
    type priority-type;
    default "4";
    description
      "Service access priority value for priority 4";
    reference
      "12.6.2.17, 6.13.1 of IEEE Std 802.1Q";
  }
  leaf priority5 {
    type priority-type;
    default "5";
    description
      "Service access priority value for priority 5";
    reference
      "12.6.2.17, 6.13.1 of IEEE Std 802.1Q";
  }
  leaf priority6 {
    type priority-type;
    default "6";
    description
      "Service access priority value for priority 6";
    reference
      "12.6.2.17, 6.13.1 of IEEE Std 802.1Q";
  }
  leaf priority7 {
    type priority-type;
    default "7";
    description
      "Service access priority value for priority 7";
    reference
      "12.6.2.17, 6.13.1 of IEEE Std 802.1Q";
  }
}
grouping traffic-class-table-grouping {
  description
    "The Traffic Class Table models the operations that can be performed  

    performed on, or can inquire about, the current contents of the Traffic Class  

Tablethe Traffic Class Table (8.6.6) for a given Port.";
  reference
    "12.6.3, 8.6.6 of IEEE Std 802.1Q";
  list traffic-class-map {
    key "priority";
    status "deprecated";
    description

```

```
"The priority index into the traffic class table. This list modeled
the Traffic Class Table incorrectly. available-traffic-class should
be a single value per port and not a list of all possible
available-traffic-class. For more information see maintenance issue
0230. The status of this object is deprecated. It is replaced by
sibling container 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";
}
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";
  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";
  }
  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";
  }
}
}
container traffic-class-table {
  description
    "The priority to traffic class mapping.";
  reference
    "8.6.6 of IEEE Std 802.1Q";
  leaf number-of-traffic-classes {
    type uint8 {
      range "1..8";
    }
    description
      "The number of egress traffic classes supported on this port. This
      object may be optionally read-only.";
    reference
      "12.6.3.1 of IEEE Std 802.1Q";
  }
  leaf priority0 {
    type traffic-class-type;
    must
      "current() < ../number-of-traffic-classes";
    description
      "The traffic class index associated with priority 0";
    reference
      "8.6.6 of IEEE Std 802.1Q";
  }
  leaf priority1 {
    type traffic-class-type;
    must
      "current() < ../number-of-traffic-classes";
    description
      "The traffic class index associated with priority 1";
    reference
      "8.6.6 of IEEE Std 802.1Q";
  }
}
```

```
leaf priority2 {
  type traffic-class-type;
  must
    "current() < ../number-of-traffic-classes";
  description
    "The traffic class index associated with priority 2";
  reference
    "8.6.6 of IEEE Std 802.1Q";
}
leaf priority3 {
  type traffic-class-type;
  must
    "current() < ../number-of-traffic-classes";
  description
    "The traffic class index associated with priority 3";
  reference
    "8.6.6 of IEEE Std 802.1Q";
}
leaf priority4 {
  type traffic-class-type;
  must
    "current() < ../number-of-traffic-classes";
  description
    "The traffic class index associated with priority 4";
  reference
    "8.6.6 of IEEE Std 802.1Q";
}
leaf priority5 {
  type traffic-class-type;
  must
    "current() < ../number-of-traffic-classes";
  description
    "The traffic class index associated with priority 5";
  reference
    "8.6.6 of IEEE Std 802.1Q";
}
leaf priority6 {
  type traffic-class-type;
  must
    "current() < ../number-of-traffic-classes";
  description
    "The traffic class index associated with priority 6";
  reference
    "8.6.6 of IEEE Std 802.1Q";
}
leaf priority7 {
  type traffic-class-type;
  must
    "current() < ../number-of-traffic-classes";
  description
    "The traffic class index associated with priority 7";
  reference
    "8.6.6 of IEEE Std 802.1Q";
}
}
}
grouping transmission-selection-table-grouping {
  description
    "The Transmission Selection Algorithm Table models the operations that
    can be performed on, or can inquire about, the current contents of the
    Transmission Selection Algorithm Table (12.20.2) for a given Port.";
  reference
    "12.20.2, 8.6.8 of IEEE Std 802.1Q";
  list transmission-selection-algorithm-map {
    key "traffic-class";
    description
      "The traffic class to index into the transmission selection table.";
    leaf traffic-class {
      type traffic-class-type;
      description
        "The traffic class of the entry.";
      reference
```

```

    "8.6.6 of IEEE Std 802.1Q";
  }
  leaf transmission-selection-algorithm {
    type identityref {
      base dot1q-types:transmission-selection-algorithm;
    }
    description
      "Transmission selection algorithm";
    reference
      "8.6.8, Table 8-6 of IEEE Std 802.1Q";
  }
}
}
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, 8.8.2 of IEEE Std 802.1Q";
  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";
    }
    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 Portr, specifying that
        specifying that a frame with a destination MAC address, and in the case of
        the case of VLAN Bridge components, VID that meets this this specification.";
        — specification.reference
        reference
        "8.8.1 of IEEE Std 802.1Q";
    }
  }
  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, 8.8.12 of IEEE Std 802.1Q";
  }
}
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";
}
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";
}
}
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 {
      — descriptionenum not-registered {
        — "Filtered, independently of any dynamic filteringdescription
        information.";Filtered, independently of any dynamic filtering
        information.";
      }
      description)
      — "containing a control element for each outbound Port,description
      "A control element for each outbound Port, specifying that a frame with a
      destination MAC address, and in that
      a frame with a destination MAC address, and in the case of VLAN Bridge
      components, VID that meets this of
      VLAN Bridge components, VID that meets thisspecification.";
```

```

    reference
      "8.8.4 of IEEE Std 802.1Q";
  }
}
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 A control element for each outbound Port, specifying that
        specifying that a frame with a destination MAC address, and in the case of
        the case of VLAN Bridge components, VID that meets this this specification.";
      — specification.";reference
      reference
        "8.8.5 of IEEE Std 802.1Q";
  }
}
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 A control element for each outbound Port, specifying that
        specifying that a frame with a destination MAC address, and in the case of
        the case of VLAN Bridge components, VID that meets this this specification.";
      — specification.";reference
      reference
        "8.8.7 of IEEE Std 802.1Q";
  }
}
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,description
        A control element for each outbound Port, specifying that a frame with a
        destination MAC address, and in that
        a frame with a destination MAC address, and in the case of VLAN Bridge
        components, VID that meets thisof
        VLAN Bridge components, VID that meets this specification.";
    reference
      "8.8.3 of IEEE Std 802.1Q";
  }
}
}

```

```
}
}
grouping bridge-port-statistics-grouping {
  description
    "Grouping of bridge port statistics.";
  reference
    "12.6.1.1.3 of IEEE Std 802.1Q";
  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, 8.6.6 of IEEE Std 802.1Q";
  }
  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
      "Item g) in 12.6.1.1.3 of IEEE Std 802.1Q";
  }
  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";
  }
  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";
  }
  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";
  }
  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";
  }
  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";
  }
  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";
  }
  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";
  }
}
grouping base-gate-control-entries {
  description
    "Grouping of gate control entries.";
  list gate-control-entry {
    key "index";
    description
      "List of gate control entries.";
    leaf index {
      type uint32;
      description
        "The index for this entry.";
    }
    leaf operation-name {
      type identityref {
        base type-of-operation;
      }
      mandatory true;
      description
        "The name (type) of the operation for this entry.";
    }
    leaf time-interval-value {
      type uint32;
      description
        "timeIntervalValue is a 32-bit unsigned integer, representing a
        number of nanoseconds. After timeIntervalValue nanoseconds have
        elapsed since the completion of the previous entry in the gate
        control list, control passes to the next entry.";
      reference
        "12.29.1.2.3, 8.6.8.4 of IEEE Std 802.1Q";
    }
  }
}
}
}

```

48.6.4 The ieee802-dot1q-bridge YANG module

Delete the YANG module in 48.6.4.

Insert the following YANG module in 48.6.4:

```
module ieee802-dot1q-bridge {
  yang-version "1.1";
  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://ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: stds-802-1-chairs@ieee.org";
  description
    "This YANG module describes the bridge Bridge configuration model for the
    following IEEE 802.1Q Bridges:
      1) Two Port MAC Relays
      2) Customer VLAN Bridges
      3) Provider Bridges.

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-0410-17-26 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE Std 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023.
      IEEE Std 802 Overview and Architecture:
      IEEE Std 802-2014.
      IEEE Std 802.1AC Media Access Control (MAC) Service Definition:
      IEEE Std 802.1AC-2016, IEEE Std 802.1AC-2016-Cor1-2018.";
  }
  revision 2022-10-29 {
    description
      "Published as part of IEEE Std 802.1Q-2022.";
    reference
      "IEEE Std 802.1Q-2022, Bridges and Bridged Networks.";
  }
}
```

IEEE Std 802.1Qcw™-2023
IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks
Amendment 36: YANG Data Models for Scheduled Traffic, Frame Preemption, and Per-Stream Filtering and Policing

```
revision 2020-11-06 {
  description
    "Published as part of IEEE Std 802.1Qcr-2020. Third version.";
  reference
    "IEEE Std 802.1Qcr-2020, Bridges and Bridged Networks - Asynchronous
    Traffic Shaping.";
}
revision 2020-06-04 {
  description
    "Published as part of IEEE Std 802.1Qcx-2020. Second version.";
  reference
    "IEEE Std 802.1Qcx-2020, Bridges and Bridged Networks - YANG Data
    Model for Connectivity Fault Management.";
}
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";
}
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-Port's 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, Clause 10 of IEEE Std 802.1Q";
}
feature port-and-protocol-based-vlan {
  description
    "A VLAN-aware bridge-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";
}
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";
}
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 Bridge Port provides an Interface-interface that is
    a Bridge Port to distinct from
    be a separate Interface from the interface provided by 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 dot1qt-types:name-type;
        description
          "A text string associated with the Bridge, of locally determined
          significance.";
        reference
          "12.4 of IEEE Std 802.1Q";
      }
      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";
      }
      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";
  }
  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";
  }
  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
      "Item 1) a in 12.4.1.5 of IEEE Std 802.1Q";
    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
        "Item 1) in 12.3 of IEEE Std 802.1Q";
    }
    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
        "Item m) in 12.3 of IEEE Std 802.1Q";
    }
    leaf address {
      type ieee:mac-address;
      description
        "Unique EUI-48 Universally Administered MAC address assigned to
        a Bridge component.";
      reference
        "8.13.8 of IEEE Std 802.1Q";
    }
  }
}
```

```

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
    "Item b\) in 12.4.1.5.4-2 of IEEE Std 802.1Q";
}
leaf ports {
  type uint16 {
    range "1..4095";
  }
  config false;
  description
    "The number of Bridge Ports associated with the Bridge
    Component.";
  reference
    "Item c) in 12.4.1.1.3 of IEEE Std 802.1Q";
}
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
    "Item b) in 12.10.1.1.3, 12.4.1.5.2 of IEEE Std 802.1Q";
  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";
  }
  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";
  }
  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";
  }
  leaf ivl-capable {
    type boolean;
    default "true";
    description
      "Independent VLAN Learning (IVL).";
    reference
      "12.4.1.5.2 of IEEE Std 802.1Q";
  }
  leaf svl-capable {
    type boolean;
    default "false";
  }
}

```

```
        description
            "Shared VLAN Learning (SVL).";
        reference
            "12.4.1.5.2 of IEEE Std 802.1Q";
    }
    leaf hybrid-capable {
        type boolean;
        default "false";
        description
            "Both IVL and SVL simultaneously.";
        reference
            "12.4.1.5.2 of IEEE Std 802.1Q";
    }
    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";
    }
    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";
    }
}
container filtering-database {
    when
        "not(derived-from-or-self(../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";
    leaf aging-time {
        type uint32 {
            range "10..1000000";
        }
        units "seconds";
        default "300";
        description
            "The timeout period in seconds for aging out
            dynamically-learned forwarding information.";
        reference
            "12.7, 8.8.3 of IEEE Std 802.1Q";
    }
    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";
    }
    leaf static-entries {
        type yang:gauge32;
        config false;
        description
            "The number of Static Filtering entries currently in the FDB.";
        reference
```



```
        "12.7, 8.8.1 of IEEE Std 802.1Q";
    }
    leaf dynamic-entries {
        type yang:gauge32;
        config false;
        description
            "The number of Dynamic Filtering entries currently in the FDB.";
        reference
            "12.7, 8.8.3 of IEEE Std 802.1Q";
    }
    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, 8.8.2 of IEEE Std 802.1Q";
    }
    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, 8.8.5 of IEEE Std 802.1Q";
    }
    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, 8.8.4 of IEEE Std 802.1Q";
    }
    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";
        }
        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";
        }
        leaf vids {
            type dot1qt-types:vid-range-type;
            description
                "The set of VLAN identifiers to which this entry applies.";
            reference
                "12.7.7 of IEEE Std 802.1Q";
        }
        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";
}
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";
  leaf database-id {
    type uint32;
    description
      "The identity of this Filtering Database.";
    reference
      "12.7.7 of IEEE Std 802.1Q";
  }
  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";
  }
  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";
  }
  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";
  }
  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";
  }
  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";
  }
  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";
    }
    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";
}
leaf vids {
  type dot1qt-types:vid-range-type;
  description
    "The set of VLAN identifiers to which this entry applies.";
  reference
    "12.7.7 of IEEE Std 802.1Q";
}
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 dot1qt-types:port-map-grouping;
}
}
container bridge-vlan {
  when
    "not(derived-from-or-self(..../bridge-type, "+
    "'two-port-mac-relay-bridge'))" {
    description
      "Applies to non TPMRs.";
  }
  description
    "The Bridge VLAN container models configuration information that
    modifymodifies, or inquireinquires about, the overall configuration of the
    BridgesBridge's VLAN resources. There is a single Bridge VLAN
    Configuration managed object per Bridge.";
  reference
    "12.10 of IEEE Std 802.1Q";
  leaf version {
    type uint16;
    config false;
    description
      "The version number supported.";
    reference
      "Item a\) of 12.10.1.1.3 of IEEE Std 802.1Q";
  }
}

```

```
leaf max-vids {
  type uint16;
  config false;
  description
    "The maximum number of VIDs supported.";
  reference
    "Item b\)1\) of 12.10.1.1.3 of IEEE Std 802.1Q";
}
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
    "Item b\)2\) of 12.10.1.1.3 of IEEE Std 802.1Q";
}
leaf protocol-template {
  if-feature "port-and-protocol-based-vlan";
  type dot1qt: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";
}
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
    "Item b\)4\) of 12.10.1.7-1.3 of IEEE Std 802.1Q";
}
list vlan {
  key "vid";
  description
    "List of VLAN related configuration nodes associated with the
    Bridge.";
  reference
    "12.10.2 of IEEE Std 802.1Q";
  leaf vid {
    type dot1qt:vlan-index-type;
    description
      "The VLAN identifier to which this entry applies.";
    reference
      "12.10.2 of IEEE Std 802.1Q";
  }
  leaf name {
    type dot1qt:name-type;
    description
      "A text string of up to 32 characters of locally determined
      significance.";
    reference
      "12.10.2 of IEEE Std 802.1Q";
  }
  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, 8.8.2 of IEEE Std 802.1Q";
  }
  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, 8.8.10 of IEEE Std 802.1Q";
  }
}
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, 6.12.3 of IEEE Std 802.1Q";
  leaf db-index {
    type uint16;
    description
      "The protocol group database index.";
  }
  leaf frame-format-type {
    type dot1qt: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";
  }
  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, rfc1042 and snap8021H, this is the 16-bit
      (2-octet) IEEE 802 Clause 9.3 EtherType field. For
      - For snapOther, this is the 40-bit (5-octet) PID. For llcOther,
      - For llcOther, this is the 2-octet IEEE 802.2 Link Service Access
      Access Point (LSAP) pair: first octet for Destination Service Access
      Access Point (DSAP) and second octet for Source Service Access
      Point (SSAP).";
    reference
      "12.10.1.7 of IEEE Std 802.1Q";
    case ethernet-rfc1042-snap8021H {
      when
        "frame-format-type = 'Ethernet' or frame-format-type = '+'
        or '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 dot1qt:ethertype-type;
        description
          "Format containing the 16-bit IEEE 802 EtherType field.";
        reference
          "9.3-2 of IEEE Std 802";
      }
    }
    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";
    }
}
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";
        }
    }
}
}
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";
}
}
list vid-to-fid-allocation {
    key "vids";
    description
        "This list allows inquiries about VID to FID allocations.";
    leaf vids {
        type dot1qt-types:vid-range-type;
        description
            "Range of VLAN identifiers.";
        reference
            "12.10.3 of IEEE Std 802.1Q";
    }
    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";
    }
}
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";
}
}
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";
  }
  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";
  }
  leaf-list vid {
    type dot1qtotypes:vlan-index-type;
    config false;
    description
      "The VLAN identifier to which this entry applies.";
    reference
      "12.7.7 of IEEE Std 802.1Q";
  }
}
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, 12.10.3.5 of IEEE Std 802.1Q";
  leaf vid {
    type dot1qtotypes:vlan-index-type;
    description
      "A list of VLAN identifier VIDs associated with a given database database
identifier identifier (i.e., FID).";
    reference
      "12.7.7 of IEEE Std 802.1Q";
  }
  leaf fid {
    type uint32;

```



```

    description
      "The Filtering Database used by this VLAN";
    reference
      "12.10.3 of IEEE Std 802.1Q";
  }
}
}
container bridge-mst {
  when
    "not(derived-from-or-self ../../bridge-type, '+'
      "'two-port-mac-relay-bridge')'" {
    description
      "Applies to non TPMRs.";
  }
  description
    "The Bridge MST container models configuration information that
    modifymodifies, or inquireinquires about, the overall configuration of the
    Bridges MST resources.";
  reference
    "12.12 of IEEE Std 802.1Q";
  leaf-list mstid {
    type dot1qtotypes: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";
    leaf fid {
      type uint32;
      description
        "The Filtering Database identifier.";
      reference
        "12.12.2 of IEEE Std 802.1Q";
    }
    leaf mstid {
      type dot1qtotypes:mstid-type;
      description
        "The MSTID to which the FID is to be allocated.";
      reference
        "12.12.2 of IEEE Std 802.1Q";
    }
  }
  list fid-to-mstid-allocation {
    key "fids";
    description
      "The FID to MSTID allocation table";
    leaf fids {
      type dot1qtotypes:vid-range-type;
      description
        "Range of FIDs.";
      reference
        "12.12.2 of IEEE Std 802.1Q";
    }
    leaf mstid {
      type dot1qtotypes:mstid-type;
      description
        "The MSTID to which the FID is allocated.";
      reference
        "12.12.2 of IEEE Std 802.1Q";
    }
  }
}
}
}
}
augment "/if:interfaces/if:interface" {
  when

```

```

    "derived-from-or-self(if:type,'ianaif:bridge') or "+
    "derived-from-or-self(if:type,'ianaif:ethernetCsmacd') or "+
    "derived-from-or-self(if:type,'ianaif:ieee8023adLag') or "+
    "derived-from-or-self(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 bridge-name {
    type leafref {
      path '/dot1q:bridges/dot1q:bridge/dot1q:name';
    }

    description
      "Used to reference configured Bridge node.";
  }
  leaf component-name {
    type leafref {
      path
        '/dot1q:bridges'+
        '/dot1q:bridge[dot1q:name=current()'+
        '/../'+
        '/bridge-name]'+
        '/dot1q:component'+
        '/dot1q:name';
    }

    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";
  }
  leaf pvid {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "/../dot1q:bridge-name]/dot1q:component[name=current()"+
      "/../dot1q:component-name]/dot1q:type !="+
      "'dot1q:d-bridge-component'" {
      description
        "Applies to non TPMRs";
    }
    type dot1qt-types:vlan-index-type;
    default "1";
    description
      "The primary (default) VID assigned to a specific Bridge Port.";
    reference
      "12.10.1, 5.4, item m) of IEEE Std 802.1Q";
  }
  leaf default-priority {
    type dot1qt-types:priority-type;
    default "0";
    description
      "The default priority assigned to a specific Bridge Port.";
    reference
      "12.6.2 of IEEE Std 802.1Q";
  }
  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, 6.9.4 of IEEE Std 802.1Q";
  uses dot1qtypes:priority-regeneration-table-grouping;
}
leaf pcpc-selection {
  type dot1qtypes:pcpc-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, 6.9.3 of IEEE Std 802.1Q";
}
container pcpc-decoding-table {
  description
    "The Priority Code Point Decoding Table parameters associated with
    a specific Bridge Port.";
  uses dot1qtypes:pcpc-decoding-table-grouping;
}
container pcpc-encoding-table {
  description
    "The Priority Code Point Encoding Table parameters associated with
    a specific Bridge Port.";
  uses dot1qtypes:pcpc-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, 6.9.3 of IEEE Std 802.1Q";
}
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 (6.9.3)
    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, 6.9.3 of IEEE Std 802.1Q";
}
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, 6.13 of IEEE Std 802.1Q";
}
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, 6.13.1 of IEEE Std 802.1Q";
    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, 8.6.6 of IEEE Std 802.1Q";
    uses dot1qtypes:traffic-class-table-grouping;
  }
  container transmission-selection-algorithm-table {
    description
      "The Transmission Selection Algorithm Table for a given Port
      assigns, for each traffic class that the Port supports, the
      transmission selection algorithm that is to be used to select
      frames for transmission from the corresponding queue. Transmission
      Selection Algorithm Tables may be managed, and allow the
      identification of vendor-specific transmission selection
      algorithms. The transmission selection algorithms are identified
      in the Transmission Selection Algorithm Table by means of integer
      identifiers.";
    reference
      "12.20.2, 8.6.8 of IEEE Std 802.1Q";
    uses dot1qtypes:transmission-selection-table-grouping;
  }
  leaf acceptable-frame {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "/../dot1q:bridge-name]/dot1q:component[name=current()"+
      "/../dot1q:component-name]/dot1q:type !="+
      "'dot1q: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, 6.9 of IEEE Std 802.1Q";
  }
  leaf enable-ingress-filtering {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "/../dot1q:bridge-name]/dot1q:component[name=current()"+
      "/../dot1q:component-name]/dot1q:type !="+
      "'dot1q: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, 8.6.2 of IEEE Std 802.1Q";
  }
  leaf enable-restricted-vlan-registration {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "/../dot1q:bridge-name]/dot1q:component[name=current()"+
      "/../dot1q:component-name]/dot1q:type !="+
      "'dot1q: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, 12.10.1.6 of IEEE Std 802.1Q";
  }
  leaf enable-vid-translation-table {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "/../dot1q:bridge-name]/dot1q:component[name=current()"+
      "/../dot1q:component-name]/dot1q:type !="+
      "'dot1q: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 not support a VID
      Translation Table.";
    reference
      "12.10.1.8, 6.9 of IEEE Std 802.1Q";
  }
  leaf enable-egress-vid-translation-table {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "/../dot1q:bridge-name]/dot1q:component[name=current()"+
      "/../dot1q:component-name]/dot1q:type !="+
      "'dot1q: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
      "12.10.1.8, 6.9 of IEEE Std 802.1Q";
  }
  list protocol-group-vid-set {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "/../dot1q:bridge-name]/dot1q:component[name=current()"+
      "/../dot1q:component-name]/dot1q:type !="+
      "'dot1q: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";
leaf group-id {
  type uint32;
  description
    "The protocol group identifier";
  reference
    "12.10.1.7 of IEEE Std 802.1Q";
}
leaf-list vid {
  type dot1qtypes:vlanid;
  description
    "The VLAN identifier to which this entry applies.";
  reference
    "12.10.2 of IEEE Std 802.1Q";
}
}
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, 6.8.2–3 of IEEE Std 802.1Q; 11.3 of IEEE Std 802.1AC";
}
leaf protocol-based-vlan-classification {
  when
    "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
    "/../dot1q:bridge-name]/dot1q:component[name=current()"+
    "/../dot1q:component-name]/dot1q:type !="+
    "'dot1q: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";
  }
  leaf max-vid-set-entries {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "/../dot1q:bridge-name]/dot1q:component[name=current()"+
      "/../dot1q:component-name]/dot1q:type !="+
      "'dot1q: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";
  }
  leaf port-number {
    type dot1qtypes:port-number-type;
    config false;
    description
      "An integer that uniquely identifies a Bridge Port.";
    reference
      "Item i) in 12.3, 17.3.2.2 of IEEE Std 802.1Q";
  }
  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, Item a) in 12.4.2-1.1.3 of IEEE Std 802.1Q";
  }
  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
      "Item c) in 12.10.1.1.3, 12.4.2 of IEEE Std 802.1Q";
  }
  leaf type-capabilities {
    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";
  }
  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";
  }
  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";
  }
  leaf media-dependent-overhead {
    type uint8;
    units "octets";
    config false;
    description
      "The portMediaDependentOverhead parameter provides the number of
      additional octets for media-dependent framing. The overhead
      includes all octets prior the first octet of the Destination
      Address field and all octets after the last octet of the frame
      check sequence.";
    reference
      "12.4.2 of IEEE Std 802.1Q";
  }
  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
        "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
        "/../../dot1q:bridge-name]/dot1q:component[name=current()"+
        "/../../dot1q:component-name]/dot1q:type !="+
        "'dot1q: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";
  }
}
list vid-translations {
  when
    "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
    "/../dot1q:bridge-name]/dot1q:component[name=current()"+
    "/../dot1q:component-name]/dot1q:type !="+
    "'dot1q: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 dot1qt-types:vlanid;
    description
      "The Local VID after translation received at the ISS or EISS.";
    reference
      "12.10.1.8, 6.9 of IEEE Std 802.1Q";
  }
  leaf relay-vid {
    type dot1qt-types:vlanid;
    description
      "The Relay VID received before translation received at ISS or
      EISS.";
    reference
      "12.10.1.8, 6.9 of IEEE Std 802.1Q";
  }
}
list egress-vid-translations {
  when
    "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
    "/../dot1q:bridge-name]/dot1q:component[name=current()"+
    "/../dot1q:component-name]/dot1q:type !="+
    "'dot1q: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 dot1qt-types:vlanid;
    description
      "The Relay VID received before translation received at ISS or
      EISS.";
    reference
      "12.10.1.9, 6.9 of IEEE Std 802.1Q";
  }
  leaf local-vid {
    type dot1qt-types:vlanid;

```

```
        description
            "The Local VID after translation received at the ISS or EISS.";
        reference
            "12.10.1.9, 6.9 of IEEE Std 802.1Q";
    }
}
}
```

48.6.6 The ieee802-dot1q-pb YANG module

Delete the YANG module in 48.6.6.

Insert the following YANG module in 48.6.6:

```
module ieee802-dot1q-pb {
  yang-version "1.1";
  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://ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: stds-802-1-chairs@ieee.org";
  description
    "This YANG module describes the bridge-Bridge configuration model for Provider
    Bridges.

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-0310-0822 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE Stds-Std 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023.";
  }
  revision 2022-01-19 {
    description
      "Published as part of IEEE Std 802.1Q-2022.";
    reference
      "IEEE Std 802.1Q-2022, Bridges and Bridged Networks.";
  }
  revision 2020-06-04 {
    description
      "Published as part of IEEE Std 802.1Qcx-2020. Second version.";
    reference
      "IEEE Std 802.1Qcx-2020, Bridges and Bridged Networks - YANG Data
      Model for Connectivity Fault Management.";
  }
  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";
  }
  list cvid-registration {
    when
      "/dot1q:bridges/dot1q:bridge[dot1q:name=current()"+
      "../dot1q:bridge-name]/dot1q:component[dot1q:name=current()"+
      "../dot1q:component-name]/dot1q:type = '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";
    }
    leaf svid {
      type dot1qtypes:vlanid;
      description
        "Service VLAN identifier.";
      reference
        "12.13.2.1 of IEEE Std 802.1Q";
    }
    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";
    }
    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";
    }
  }
}
list service-priority-regeneration {
  when
    "/dot1q:bridges/dot1q:bridge[dot1q:name=current() "+
    "../dot1q:bridge-name]/dot1q:component[dot1q:name=current() "+
    "../dot1q:component-name]/dot1q:type = '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";
  }
  container priority-regeneration {
    description
      "Contains Service Priority Regeneration table nodal information.";
    reference
      "12.13.2.6 of IEEE Std 802.1Q";
    uses dot1qtypes:priority-regeneration-table-grouping;
  }
}
list rcap-internal-interface {
  when
    "/dot1q:bridges/dot1q:bridge[dot1q:name=current() "+
    "../dot1q:bridge-name]/dot1q:component[dot1q:name=current() "+
    "../dot1q:component-name]/dot1q:type = '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";
  }
  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";
  }
  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";
  }
  leaf internal-interface-type {
    type enumeration {
      enum port-based-rcsi {
        description
          "Port-based RCSI";
      }
      enum c-tagged-rcsi {
        description
          "C-tagged RCSI";
      }
      enum pnp {

```

IEEE Std 802.1Qcw™-2023
IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks
Amendment 36: YANG Data Models for Scheduled Traffic, Frame Preemption, and Per-Stream Filtering and Policing

```
        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";
}
}
}
```

Insert 48.6.17 after 48.6.16 (inserted by IEEE Std 801.Qcz-2023) as follows:

48.6.17 The ieee802-dot1q-sched YANG module

```
module ieee802-dot1q-sched {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-sched;
  prefix sched;
  import ietf-yang-types {
    prefix yang;
  }
  import ieee802-types {
    prefix ieee802;
  }
  import ieee802-dot1q-types {
    prefix dot1q-types;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: STDS-802-1-CHAIRS@IEEE.ORG";
  description
    "This module provides for management of IEEE Std 802.1Q Bridges that
    support Scheduled Traffic Enhancements.

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-03-10-17-22 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE Std Std 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023.";
  }
  feature scheduled-traffic {
    description
      "Enhancements for Scheduled Traffic supported.";
    reference
      "IEEE Std 802.1Q";
  }
  identity set-gate-states {
    base dot1q-types:type-of-operation;
    description
      "Operation to set the gate states.";
  }
  identity set-and-hold-mac {
    base dot1q-types:type-of-operation;
    description
      "Operation to set and hold MAC.";
  }
  identity set-and-release-mac {
    base dot1q-types:type-of-operation;
    description
      "Operation to set and release MAC.";
  }
  grouping sched-gate-control-entries {
```



```

description
  "A GateControlEntry consists of an operation name, followed by up to 2
  parameters associated with the operation. The first parameter is a
  gateStatesValue; the second parameter is a timeIntervalValue";
uses dot1q-types:base-gate-control-entries {
  refine "gate-control-entry/operation-name" {
    must
      "(. = 'sched:set-gate-states') or (. = '+'
      'sched:set-and-hold-mac') or (. = 'sched:set-and-release-mac')";
  }
  refine "gate-control-entry/time-interval-value" {
    must
      "(. <= ../../../../supported-interval-max )";
  }
  augment "gate-control-entry" {
    description
      "Augment gate-control-entry from base-gate-control-entries with
      parameter gate-states-value.";
    leaf gate-states-value {
      type uint8;
      mandatory true;
      description
        "gateStatesValue is the gate states for this entry for the Port.
        The gates are immediately set to the states in gateStatesValue
        when this entry executes. The bits of the octet represent the
        gate states for the corresponding traffic classes; the
        most-significant bit corresponds to traffic class 7, the
        least-significant bit to traffic class 0. A bit value of 0
        indicates closed; a bit value of 1 indicates open.";
      reference
        "12.29.1.2.2 and 8.6.8.4 of IEEE Std 802.1Q.";
    }
  }
}
}
grouping sched-parameters {
  description
    "sched-parameters comprises all of the parameters associated with
    Scheduled Traffic configuration.";
  container gate-parameter-table {
    description
      "A table that contains the per-port manageable parameters for
      traffic scheduling. For a given Port, an entry in the table exists.
      All writable objects in this table must be persistent over power up
      restart/reboot.";
    reference
      "12.29.1, 8.6.8.4 and 8.6.9 of IEEE Std 802.1Q";
    list queue-max-sdu-table {
      key "traffic-class";
      description
        "A list containing a set of max SDU parameters, one for each
        traffic class. All writable objects in this table must be
        persistent over power up restart/reboot.";
      reference
        "12.29.1.1, 8.6.8.4 and 8.6.9 of IEEE Std 802.1Q";
      leaf traffic-class {
        type dot1q-types:traffic-class-type;
        description
          "Traffic class";
      }
      leaf queue-max-sdu {
        type uint32;
        default "0";
        description
          "The value of the queueMaxSDU parameter for the traffic class. A
          value of 0 is interpreted as the max SDU size supported by the
          underlying MAC. The value must be retained across
          reinitializations of the management system.";
        reference
          "12.29.1.1.1, 8.6.8.4 and 8.6.9 of IEEE Std 802.1Q.";
      }
    }
    leaf transmission-overrun {

```

```
    type yang:counter64;
    default "0";
    config false;
    description
      "A counter of transmission overrun events, where a PDU is still
      being transmitted by a MAC at the time when the transmission
      gate for the queue closed.";
    reference
      "12.29.1.1.2, 8.6.8.4 and 8.6.9 of IEEE Std 802.1Q.";
  }
}
leaf gate-enabled {
  type boolean;
  default "false";
  description
    "The GateEnabled parameter determines whether traffic scheduling
    is active (true) or inactive (false). The value must be retained
    across reinitializations of the management system.";
  reference
    "12.29.1, 8.6.8.2 and 8.6.9.4.14 of IEEE Std 802.1Q.";
}
leaf admin-gate-states {
  type uint8;
  default "255";
  description
    "AdminGateStates is the administrative value of the initial gate
    states for the Port. The bits of the octet represent the gate
    states for the corresponding traffic classes; the most-significant
    bit corresponds to traffic class 7, the least-significant bit to
    traffic class 0. A bit value of 0 indicates closed; a bit value of
    1 indicates open. The value must be retained across
    reinitializations of the management system.";
  reference
    "12.29.1 and 8.6.9.4.5 of IEEE Std 802.1Q.";
}
leaf oper-gate-states {
  type uint8;
  config false;
  description
    "OperGateStates is the operational value of the current gate
    states for the Port. The bits of the octet represent the gate
    states for the corresponding traffic classes; the most-significant
    bit corresponds to traffic class 7, the least-significant bit to
    traffic class 0. A bit value of 0 indicates closed; a bit value of
    1 indicates open.";
  reference
    "12.29.1 and 8.6.9.4.21 of IEEE Std 802.1Q.";
}
container admin-control-list {
  must
    "(count(/gate-control-entry) <= ../supported-list-max)" {
      error-message
        "Number of elements in admin-control-list must not be greater"+
        "than supported-list-max";
    }
  description
    "AdminControlList is the administrative value of the gate control
    list for the Port. The value must be retained across
    reinitializations of the management system.";
  reference
    "12.29.1.2, 8.6.8.4 and 8.6.9.4.2 of IEEE Std 802.1Q.";
  uses sched-gate-control-entries;
}
container oper-control-list {
  must
    "(count(/gate-control-entry) <= ../supported-list-max)" {
      error-message
        "Number of elements in oper-control-list must not be greater"+
        "than supported-list-max";
    }
  config false;
  description
```

```
    "OperControlList is the operational value of the gate control list
    for the Port.";
  reference
    "12.29.1.2, 8.6.8.4 and 8.6.9.4.18 of IEEE Std 802.1Q.";
  uses sched-gate-control-entries;
}
container admin-cycle-time {
  must
    "(./numerator div ./denominator <= "+
    "../supported-cycle-max/numerator div "+
    "../supported-cycle-max/denominator )" {
    error-message
      "admin-cycle-time must not be greater than supported-cycle-max";
    }
  description
    "AdminCycleTime specifies the administrative value of the gating
    cycle time for the Port. AdminCycleTime is a rational number of
    seconds, defined by an integer numerator and an integer
    denominator. The value must be retained across reinitializations
    of the management system.";
  reference
    "12.29.1, 8.6.8.4 and 8.6.9.4.3 of IEEE Std 802.1Q.";
  uses ieee802:rational-grouping;
}
container oper-cycle-time {
  must
    "(./numerator div ./denominator <= "+
    "../supported-cycle-max/numerator div "+
    "../supported-cycle-max/denominator )" {
    error-message
      "oper-cycle-time must not be greater than supported-cycle-max";
    }
  config false;
  description
    "OperCycleTime specifies the operational value of the gating cycle
    time for the Port. OperCycleTime is a rational number of seconds,
    defined by an integer numerator and an integer denominator.";
  reference
    "12.29.1, 8.6.8.4 and 8.6.9.4.19 of IEEE Std 802.1Q.";
  uses ieee802:rational-grouping;
}
leaf admin-cycle-time-extension {
  type uint32;
  units "nanoseconds";
  description
    "An unsigned integer number of nanoseconds, defining the maximum
    amount of time by which the gating cycle for the Port is permitted
    to be extended when a new cycle configuration is being installed.
    This is the administrative value. The value must be retained
    across reinitializations of the management system.";
  reference
    "12.29.1 and 8.6.9.4.4 of IEEE Std 802.1Q.";
}
leaf oper-cycle-time-extension {
  type uint32;
  units "nanoseconds";
  config false;
  description
    "An unsigned integer number of nanoseconds, defining the maximum
    amount of time by which the gating cycle for the Port is permitted
    to be extended when a new cycle configuration is being installed.
    This is the operational value.";
  reference
    "12.29.1 and 8.6.9.4.20 of IEEE Std 802.1Q.";
}
container admin-base-time {
  description
    "The administrative value of the base time at which gating cycles
    begin, expressed as an IEEE 1588 precision time protocol (PTP)
    timescale. The value must be retained across reinitializations of
    the management system.";
  reference
```

```
    "12.29.1 and 8.6.9.4.1 of IEEE Std 802.1Q.";
    uses ieee802:ptp-time-grouping;
}
container oper-base-time {
    config false;
    description
        "The operational value of the base time at which gating cycles
        begin, expressed as an IEEE 1588 precision time protocol (PTP)
        timescale.";
    reference
        "12.29.1 and 8.6.9.4.17 of IEEE Std 802.1Q.";
    uses ieee802:ptp-time-grouping;
}
leaf config-change {
    type boolean;
    description
        "The ConfigChange parameter signals the start of a configuration
        change when it is set to TRUE, indicating that the administrative
        parameters for the Port are ready to be copied into their
        corresponding operational parameters. This should only be done
        when the various administrative parameters are all set to
        appropriate values.";
    reference
        "12.29.1 and 8.6.9.4.7 of IEEE Std 802.1Q.";
}
container config-change-time {
    config false;
    description
        "The time at which the next config change is scheduled to occur.";
    reference
        "12.29.1 and 8.6.9.4.9 of IEEE Std 802.1Q.";
    uses ieee802:ptp-time-grouping;
}
leaf tick-granularity {
    type uint32;
    config false;
    description
        "The granularity of the cycle time clock, represented as an
        unsigned number of tenths of nanoseconds. The value must be
        retained across reinitializations of the management system.";
    reference
        "12.29.1 of IEEE Std 802.1Q";
}
container current-time {
    config false;
    description
        "The current time as maintained by the local system.";
    reference
        "12.29.1 and 8.6.9.4.10 of IEEE Std 802.1Q.";
    uses ieee802:ptp-time-grouping;
}
leaf config-pending {
    type boolean;
    config false;
    description
        "The value of the ConfigPending state machine variable. The value
        is TRUE if a configuration change is in progress but has not yet
        completed.";
    reference
        "12.29.1 and 8.6.9.4.8 of IEEE Std 802.1Q.";
}
leaf config-change-error {
    type yang:counter64;
    config false;
    description
        "A counter of the number of times that a re-configuration of the
        traffic schedule has been requested with the old schedule still
        running and the requested base time was in the past.";
    reference
        "12.29.1 and 8.6.9.3.1 of IEEE Std 802.1Q.";
}
leaf supported-list-max {
```

```
    type uint32;
    description
      "The maximum value supported by this Port for the
      AdminControlListLength and OperControlListLength parameters. It is
      available for use by schedule computation software to determine
      the port's control list capacity prior to computation. The object
      may optionally be read-only.";
    reference
      "12.29.1.5 of IEEE Std 802.1Q";
  }
  container supported-cycle-max {
    description
      "The maximum value supported by this Port of the AdminCycleTime
      and OperCycleTime parameters. The object may optionally be
      read-only.";
    reference
      "12.29.1.6 of IEEE Std 802.1Qcw";
    uses ieee802:rational-grouping;
  }
  leaf supported-interval-max {
    type uint32;
    description
      "The maximum value supported by this Port of the TimeIntervalValue
      parameter. The object may optionally be read-only.";
    reference
      "12.29.1.7 of IEEE Std 802.1Qcw";
  }
}
}
```

Insert 48.6.18 after 48.6.17 as follows:

48.6.18 The ieee802-dot1q-sched-bridge YANG module

```
module ieee802-dot1q-sched-bridge {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-sched-bridge;
  prefix sched-bridge;
  import ietf-interfaces {
    prefix if;
  }
  import ieee802-dot1q-bridge {
    prefix dot1q;
  }
  import ieee802-dot1q-sched {
    prefix sched;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: STDS-802-1-CHAIRS@IEEE.ORG";
  description
    "This module provides for management of IEEE Std 802.1Q Bridges that
    support Scheduled Traffic Enhancements.

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-03-10-08-26 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE Std Std 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023."*;
  }
  augment "/if:interfaces/if:interface/dot1q:bridge-port" {
    description
      "Augment bridge-port with Scheduled Traffic configuration.";
    uses sched:sched-parameters;
  }
}
```

Insert 48.6.19 after 48.6.18 as follows:

48.6.19 The ieee802-dot1q-preemption YANG module

```
module ieee802-dot1q-preemption {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-preemption;
  prefix preempt;
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: STDS-802-1-CHAIRS@IEEE.ORG";
  description
    "This module provides for management of IEEE Std 802.1Q Bridges that
    support Frame Preemptionframe preemption.

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-0310-08-26 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE StdStd 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023.";
  }
  feature frame-preemption {
    description
      "Frame preemption supported.";
    reference
      "IEEE Std 802.1Q";
  }
  typedef frame-preemption-status-enum {
    type enumeration {
      enum express {
        description
          "Frames queued for the priority are to be transmitted using the
          express service for the Port.";
      }
      enum preemptable {
        description
          "Frames queued for the priority are to be transmitted using the
          preemptable service for the Port.";
      }
    }
    description
      "Frame preemption status of each priority.";
  }
  grouping preemption-parameters {
    description
      "preemption-parameters comprises all of the parameters associated with
      Frame Preemptionframe preemption configuration.";
    container frame-preemption-parameters {
      description
        "A table containing a set of frame preemption parameters, one for
        each Port. All writeable objects in this table must be persistent
        over power up restart/reboot.";
    }
  }
}
```

```
reference
  "12.30.1 of IEEE Std 802.1Q";
container frame-preemption-status-table {
  description
    "The framePreemptionStatusTable consists of 8 frame-preemption
    status parameter leafs, one per priority";
  reference
    "12.30.1.1 of IEEE Std 802.1Q";
  leaf priority0 {
    type frame-preemption-status-enum;
    default "express";
    description
      "The value of the frame-preemption status parameter for the
      priority 0. The default value of the parameter is express (1).
      The value of this object MUST be retained across
      reinitializations of the management system.";
    reference
      "12.30.1.1.1 of IEEE Std 802.1Q";
  }
  leaf priority1 {
    type frame-preemption-status-enum;
    default "express";
    description
      "The value of the frame-preemption status parameter for the
      priority 1. The default value of the parameter is express (1).
      The value of this object MUST be retained across
      reinitializations of the management system.";
    reference
      "12.30.1.1.1 of IEEE Std 802.1Q";
  }
  leaf priority2 {
    type frame-preemption-status-enum;
    default "express";
    description
      "The value of the frame-preemption status parameter for the
      priority 2. The default value of the parameter is express (1).
      The value of this object MUST be retained across
      reinitializations of the management system.";
    reference
      "12.30.1.1.1 of IEEE Std 802.1Q";
  }
  leaf priority3 {
    type frame-preemption-status-enum;
    default "express";
    description
      "The value of the frame-preemption status parameter for the
      priority 3. The default value of the parameter is express (1).
      The value of this object MUST be retained across
      reinitializations of the management system.";
    reference
      "12.30.1.1.1 of IEEE Std 802.1Q";
  }
  leaf priority4 {
    type frame-preemption-status-enum;
    default "express";
    description
      "The value of the frame-preemption-status parameter for the
      priority 4. The default value of the parameter is express (1).
      The value of this object MUST be retained across
      reinitializations of the management system.";
    reference
      "12.30.1.1.1 of IEEE Std 802.1Q";
  }
  leaf priority5 {
    type frame-preemption-status-enum;
    default "express";
    description
      "The value of the frame-preemption status parameter for the
      priority 5. The default value of the parameter is express (1).
      The value of this object MUST be retained across
      reinitializations of the management system.";
    reference
```



```
    "12.30.1.1.1 of IEEE Std 802.1Q";
  }
  leaf priority6 {
    type frame-preemption-status-enum;
    default "express";
    description
      "The value of the frame-preemption status parameter for the
       priority 6. The default value of the parameter is express (1).
       The value of this object MUST be retained across
       reinitializations of the management system.";
    reference
      "12.30.1.1.1 of IEEE Std 802.1Q";
  }
  leaf priority7 {
    type frame-preemption-status-enum;
    default "express";
    description
      "The value of the frame-preemption status parameter for the
       priority 7. The default value of the parameter is express (1).
       The value of this object MUST be retained across
       reinitializations of the management system.";
    reference
      "12.30.1.1.1 of IEEE Std 802.1Q";
  }
}
leaf hold-advance {
  type uint32;
  units "nanoseconds";
  config false;
  description
    "The value of the holdAdvance parameter for the Port in
     nanoseconds. There is no default value; the holdAdvance is a
     property of the underlying MAC.";
  reference
    "12.30.1.2 of IEEE Std 802.1Q";
}
leaf release-advance {
  type uint32;
  units "nanoseconds";
  config false;
  description
    "The value of the releaseAdvance parameter for the Port in
     nanoseconds. There is no default value; the releaseAdvance is a
     property of the underlying MAC.";
  reference
    "12.30.1.3 of IEEE Std 802.1Q";
}
leaf preemption-active {
  type boolean;
  config false;
  description
    "TRUE if preemption is both supported by the MAC and currently
     active.";
  reference
    "12.30.1.4 of IEEE Std 802.1Q";
}
leaf hold-request {
  type enumeration {
    enum hold {
      value 1;
      description
        "A hold request has been issued to the MAC.";
    }
    enum release {
      value 2;
      description
        "A release request has been issued to the MAC.";
    }
  }
  config false;
  description
    "The value of the holdRequest parameter for the Port, either hold
```

```
        (1) or release (2). The value of this object is release (2) on  
        system initialization.";
    reference  
        "12.30.1.5 of IEEE Std 802.1Q";
    }  
}  
}
```

Insert 48.6.20 after 48.6.19 as follows:

48.6.20 The ieee802-dot1q-preemption-bridge YANG module

```
module ieee802-dot1q-preemption-bridge {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-preemption-bridge;
  prefix preempt-bridge;
  import ietf-interfaces {
    prefix if;
  }
  import ieee802-dot1q-bridge {
    prefix dot1q;
  }
  import ieee802-dot1q-preemption {
    prefix preempt;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: STDS-802-1-CHAIRS@IEEE.ORG";
  description
    "This module provides for management of IEEE Std 802.1Q Bridges that
    support Frame Preemptionframe\_preemption."

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-03-10-08-26 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE StdStd 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023."*;
  }
  feature frame-preemption {
    description
      "Frame preemption supported.";
    reference
      "IEEE Std 802.1Q";
  }
  augment "/if:interfaces/if:interface/dot1q:bridge-port" {
    if-feature "frame-preemption";
    description
      "Augment bridge-port with Frame Preemptionframe\_preemption configuration.";
    uses preempt:preemption-parameters;
  }
}
```

Insert 48.6.21 after 48.6.20 as follows:

48.6.21 The ieee802-dot1q-psfp YANG module

```
module ieee802-dot1q-psfp {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-psfp;
  prefix psfp;
  import ietf-yang-types {
    prefix yang;
  }
  import ieee802-types {
    prefix ieee802;
  }
  import ieee802-dot1q-types {
    prefix dot1q-types;
  }
  import ieee802-dot1q-stream-filters-gates {
    prefix sfsg;
  }
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: STDS-802-1-CHAIRS@IEEE.ORG";
  description
    "This module provides management of IEEE Std 802.1Q bridge components thatBridge components that support Per-Per-Stream Filtering and Policing (PSFP).

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-0310-17-26 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE StdStd 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023."7;
    }
  feature psfp {
    description
      "Per-Per-Stream Filtering and Policing supported.";
    reference
      "IEEE Std 802.1Q";
  }
  identity set-gate-and-ipv {
    base dot1q-types:type-of-operation;
    description
      "The StreamGateState parameter specifies a desired state, open or
      closed, for the stream gate, and the IPV parameter specifies a desired
      value of the IPV associated with the stream. On execution, the
      StreamGateState and IPV parameter values are used to set the
      operational values of the stream gate state and internal priority
      specification parameters for the stream. After TimeInterval ticks hashave
      elapsed since the completion of the previous stream gate control
      operation in the stream gate control list, control passes to the next
      stream gate control operation. The optional IntervalOctetMax parameter
      specifies the maximum number of MSDU octets that are permitted to pass
```

```

    the gate during the specified TimeInterval. If the IntervalOctetMax
    parameter is omitted, there is no limit on the number of octets that
    can pass the gate.";
  reference
    "8.6.5.4 of IEEE Std 802.1Q";
}
grouping stream-gate-control-entries {
  description
    "A GateControlEntry consists of an operation name, followed by three
    mandatory parameters and one optional parameter associated with the
    operation. The first parameter is a gateStatesValue; the second
    parameter is an IPV value; the third parameter is a TimeIntervalValue;
    and the fourth parameter is an IntervalOctetMaxValue.
    IntervalOctetMaxValue is optional.";
  uses dot1q-types:base-gate-control-entries {
    refine "gate-control-entry/operation-name" {
      must
        ". = 'psfp:set-gate-and-ipv'";
    }
    refine "gate-control-entry/time-interval-value" {
      must
        "(. <= ../../../supported-interval-max)";
    }
  }
  augment "gate-control-entry" {
    description
      "Augment gate-control-entry from base-gate-control-entries with
      parameters gate-state-value, ipv-spec and interval-octet-max.";
    leaf gate-state-value {
      type sfsg:gate-state-value-type;
      mandatory true;
      description
        "The PSFPGateStatesValue indicates the desired gate state, open
        or closed, for the stream gate.";
      reference
        "12.31.3.2.1 of IEEE Std 802.1Q";
    }
    leaf ipv-spec {
      type sfsg:ipv-spec-type;
      mandatory true;
      description
        "The IPV value indicates the IPV to be associated with frames
        that pass the gate.";
      reference
        "12.31.3.2.3 of IEEE Std 802.1Q";
    }
    leaf interval-octet-max {
      type uint32;
      description
        "An unsigned integer, denoting a IntervalOctetMax in MSDU
        octets. If this parameter is omitted, there is no maximum.";
      reference
        "12.31.3.2.5 of IEEE Std 802.1Qcw1Q";
    }
  }
}
}
}
grouping psfp-parameters {
  description
    "psfp-parameters comprises all of the parameters associated with
    Per-Stream Filtering and Policing configuration.";
  container flow-meters {
    description
      "This container comprises all flow meter related nodes.";
    list flow-meter-instance-table {
      must
        "(count(.) <= ../max-flow-meter-instances)" {
          error-message
            "Number of elements in flow-meter-instance-table must not be"+
            "greater than max-flow-meter-instances.";
        }
      key "flow-meter-instance-id";
      description

```

```
"There is one Flow Meter Instance Table per Bridge component. Each
table row contains a set of parameters that defines a single Flow
Meter Instance. 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 flow meters.";
reference
  "12.31.4 of IEEE Std 802.1Q";
leaf flow-meter-instance-id {
  type uint32;
  mandatory true;
  description
    "An integer table index that allows the Flow Meter to be
    referenced from Stream Filter Instance Table entries.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
leaf committed-information-rate {
  type uint64;
  units "bits/second";
  mandatory true;
  description
    "Committed information rate (CIR), in bits per second.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
leaf committed-burst-size {
  type uint32;
  units "octets";
  mandatory true;
  description
    "Committed burst size (CBS), in octets.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
leaf excess-information-rate {
  type uint64;
  units "bits/second";
  mandatory true;
  description
    "Excess Information Rate (EIR), in bits per second.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
leaf excess-burst-size {
  type uint32;
  units "octets";
  mandatory true;
  description
    "Excess burst size (EBS) per bandwidth profile flow, in octets.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
leaf coupling-flag {
  type enumeration {
    enum zero {
      value 0;
      description
        "Uncoupled";
    }
    enum one {
      value 1;
      description
        "Coupled";
    }
  }
  mandatory true;
  description
    "Coupling flag (CF), which takes the value 0 or 1.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
```

```

}
leaf color-mode {
  type enumeration {
    enum color-blind {
      description
        "Color-blind (i.e., the color of a frame is ignored by the
        bandwidth profile algorithm).";
    }
    enum color-aware {
      description
        "Color-aware (i.e., the color of a frame is considered by
        the bandwidth profile algorithm).";
    }
  }
  mandatory true;
  description
    "Color mode (CM), which takes the value color-blind or
    color-aware, as specified in Bandwidth Profile Parameters and
    Algorithm in MEF 10.3.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
leaf drop-on-yellow {
  type boolean;
  mandatory true;
  description
    "DropOnYellow, which takes the value TRUE or FALSE. A value of
    TRUE indicates that yellow frames are dropped (i.e., discarded);
    a value of FALSE indicates that yellow frames will have the
    drop_eligible parameter set to TRUE.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
leaf mark-all-frames-red-enable {
  type boolean;
  default "false";
  description
    "MarkAllFramesRedEnable, which takes the value TRUE or FALSE. A
    value of TRUE indicates that the MarkAllFramesRed function is
    enabled; a value of FALSE indicates that the MarkAllFramesRed
    function is disabled. The default value ofof MarkAllFramesRedEnable
    MarkAllFramesRedEnable is FALSE.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
leaf mark-all-frames-red {
  type boolean;
  default "false";
  description
    "MarkAllFramesRed, which takes the value TRUE or FALSE. If
    MarkAllFramesRedEnable is TRUE, a value of TRUE in
    MarkAllFramesRed indicates that all frames are dropped (i.e.,
    discarded). If MarkAllFramesRed is FalseFALSE, it has no effect. The
    default value of MarkAllFramesRed is FALSE; if the operation of
    the flow meter causes any frame to be discarded, then
    MarkAllFramesRed is set TRUE.";
  reference
    "8.6.5.5 of IEEE Std 802.1Q";
}
}
leaf max-flow-meter-instances {
  type uint32;
  description
    "The maximum number of Flow Meter instances supported by this
    Bridge component. The object may optionally be read-only.";
  reference
    "12.31.1.3 of IEEE Std 802.1Q";
}
}
uses sfsg:sfsg-parameters {
  augment "stream-filters/stream-filter-instance-table" {
    description

```

```
"Augment the Bridge component Stream filter by frame counters.";
leaf matching-frames-count {
  type yang:counter64;
  config false;
  description
    "A count of frames matching both the stream_handle and priority
    specifications.";
  reference
    "8.6.5.3 of IEEE Std 802.1Q";
}
leaf passing-frames-count {
  type yang:counter64;
  config false;
  description
    "A count of frames that passed the stream gate.";
  reference
    "8.6.5.3 of IEEE Std 802.1Q";
}
leaf not-passing-frames-count {
  type yang:counter64;
  config false;
  description
    "A count of frames that did not pass the stream gate.";
  reference
    "8.6.5.3 of IEEE Std 802.1Q";
}
leaf red-frames-count {
  type yang:counter64;
  config false;
  description
    "A count of frames that were discarded as a result of the
    operation of the flow meter.";
  reference
    "8.6.5.3 of IEEE Std 802.1Q";
}
leaf passing-sdu-count {
  type yang:counter64;
  config false;
  description
    "A count of frames that passed the Maximum SDU size filter.";
  reference
    "8.6.5.3 of IEEE Std 802.1Q";
}
leaf not-passing-sdu-count {
  type yang:counter64;
  config false;
  description
    "A count of frames that did not pass the Maximum SDU size
    filter.";
  reference
    "8.6.5.3 of IEEE Std 802.1Q";
}
leaf flow-meter-ref {
  type leafref {
    path
      '..'+
      '/..' +
      '/..' +
      '/flow-meters'+
      '/flow-meter-instance-table'+
      '/flow-meter-instance-id';
  }
  description
    "Flow meter instance that is associated with the stream filter.";
  reference
    "8.6.5.5 and 12.31.2.6 of IEEE Std 802.1Q.";
}
leaf flow-meter-enable {
  type boolean;
  description
    "If it is set to TRUE, flow-meter-instance-id identifies the
    flow meter instance associated with the stream filter. If it is
```



```

    set to FALSE, no flow meter instance is associated with the
    stream filter.";
  reference
    "8.6.5.5 and 12.31.2.6 of IEEE Std 802.1Q.";
}
}
augment "stream-gates" {
  description
    "Augment the Bridge component Stream Gates by maximum control list
    limits, as used for PTP-controlled open and close transitions";
  leaf supported-list-max {
    type uint32;
    description
      "The maximum value supported by this Bridge component of the
      AdminControlListLength and OperControlListLength parameters. It
      is available for use by schedule computation software to
      determine the Bridge component's control list capacity prior to
      computation. The object may optionally be read-only.";
    reference
      "12.31.1.4 of IEEE Std 802.1Q";
  }
  container supported-cycle-max {
    description
      "The maximum value supported by this Bridge component of the
      AdminCycleTime and OperCycleTime parameters. The object may
      optionally be read-only.";
    reference
      "12.31.1.5-7 of IEEE Std 802.1Qcw1Q";
    uses ieee802:rational-grouping;
  }
  leaf supported-interval-max {
    type uint32;
    description
      "The maximum value supported by this Bridge component of the
      TimeIntervalValue parameter. The object may optionally be
      read-only.";
    reference
      "12.31.1.6-8 of IEEE Std 802.1Qcw1Q";
  }
}
augment "stream-gates/stream-gate-instance-table" {
  description
    "Augment the Bridge component Stream Gate instances by
    a) operational Operational gate states
    b) operational Operational IPV values
    c) PTP controlled open and close transitions
    d) Management for PTP controlled open and close transitions";
  leaf oper-gate-state {
    type sfsg:gate-state-value-type;
    config false;
    description
      "The current operational state of the gate. The operational
      state of the gate is set by the List Execute state machine, and
      its initial value is determined by the value of the
      administrative state of the gate.";
    reference
      "12.31.3.2.1 and 8.6.10.5 of IEEE Std 802.1Q-";
  }
  leaf oper-ipv {
    type sfsg:ipv-spec-type;
    description
      "The operational internal priority value specification.";
    reference
      "12.31.3.3, 8.6.10.7-7, and 8.6.5.4 of IEEE Std 802.1Q-";
  }
  container admin-control-list {
    must
      "(count(/gate-control-entry) <= ../supported-list-max)" {
        error-message
          "Number of elements in admin-control-list must not be"+
          "greater than supported-list-max";
      }
  }
}

```

```
description
  "The AdminControlList and OperControlList are ordered lists
  containing AdminControlListLength or OperControlListLength
  entries, respectively. Each entry represents a gate operation.
  Each entry in the list is structured as a GateControlEntry.";
reference
  "12.31.3.2, 12.31.3.2.2-2, and 8.6.9.4.2 of IEEE Std 802.1Q-";
  uses stream-gate-control-entries;
}
container oper-control-list {
  must
    "(count(/gate-control-entry) <= ../supported-list-max)" {
      error-message
        "Number of elements in oper-control-list must not be greater"+
        "than supported-list-max";
    }
  config false;
  description
    "The AdminControlList and OperControlList are ordered lists
    containing AdminControlListLength or OperControlListLength
    entries, respectively. Each entry represents a gate operation.
    Each entry in the list is structured as a GateControlEntry.";
  reference
    "12.31.3.2, 12.31.3.2.2 and 8.6.9.4.18 of IEEE Std 802.1Q-";
    uses stream-gate-control-entries;
}
container admin-cycle-time {
  must
    "(./numerator div ./denominator <= "+
    "../supported-cycle-max/numerator div "+
    "../supported-cycle-max/denominator)" {
      error-message
        "admin-cycle-time must not be greater than"+
        "supported-cycle-max";
    }
  description
    "The administrative value of the gating cycle for the Port. This
    value can be changed by management, and is used by the List
    Config state machine to set the value of OperCycleTime. The
    AdminCycleTime variable is a rational number of seconds, defined
    by an integer numerator and an integer denominator.";
  reference
    "12.29.1.3 and 8.6.9.4.3 of IEEE Std 802.1Q-";
    uses ieee802:rational-grouping;
}
container oper-cycle-time {
  must
    "(./numerator div ./denominator <= "+
    "../supported-cycle-max/numerator div "+
    "../supported-cycle-max/denominator)" {
      error-message
        "oper-cycle-time must not be greater than"+
        "supported-cycle-max";
    }
  config false;
  description
    "The operational value of the gating cycle for the Port. This
    variable is set dynamically from the AdminCycleTime variable
    under the control of the List Config state machine OperCycleTime
    is used by the Cycle Timer state machine to enforce the cycle
    time for the Port. The OperCycleTime variable is a rational
    number of seconds, defined by an integer numerator and an
    integer denominator.";
  reference
    "12.29.1.3 and 8.6.9.4.19 of IEEE Std 802.1Q-";
    uses ieee802:rational-grouping;
}
leaf admin-cycle-time-extension {
  type uint32;
  units "nanoseconds";
  description
    "An integer number of nanoseconds, defining the maximum amount
```

```

    of time by which the gating cycle for the Port is permitted to
    be extended when a new cycle configuration is being installed.
    This administrative value can be changed by management, and is
    used by the List Config state machine to set the value of
    OperCycleTimeExtension.";
  reference
    "8.6.9.4.4 of IEEE Std 802.1Q";
}
leaf oper-cycle-time-extension {
  type uint32;
  units "nanoseconds";
  config false;
  description
    "An integer number of nanoseconds, defining the maximum amount
    of time by which the gating cycle for the Port is permitted to
    be extended when a new cycle configuration is installed. This
    operational value is set by the List Config state machine to the
    value of AdminCycleTimeExtension. The value of
    OperCycleTimeExtension is used by the SetCycleStartTime()
    procedure.";
  reference
    "8.6.9.4.20 of IEEE Std 802.1Q";
}
container admin-base-time {
  description
    "The administrative value of base time, expressed as an IEEE
    1588 precision time protocol (PTP) timescale (see IEEE Std
    802.1AS-2011). This value can be changed by management, and is
    used by the List Config state machine to set the value of
    OperBaseTime.";
  reference
    "12.29.1.4 and 8.6.9.4.1 of IEEE Std 802.1Q-";
  uses ieee802:ptp-time-grouping;
}
container oper-base-time {
  config false;
  description
    "The operational value of base time, expressed as a PTP
    timescale (see IEEE Std 802.1AS-2011). This variable is used by
    the List Config state machine.";
  reference
    "12.29.1.4 and 8.6.9.4.17 of IEEE Std 802.1Q-";
  uses ieee802:ptp-time-grouping;
}
leaf config-change {
  type boolean;
  description
    "A Boolean variable that acts as a start signal to the List
    Config state machine that the administrative variable values for
    the Port are ready to be copied into their corresponding
    operational variables. This variable is set TRUE by management
    and is set FALSE by the List Config state machine.";
  reference
    "8.6.9.4.7 of IEEE Std 802.1Q";
}
container config-change-time {
  config false;
  description
    "The time at which the administrative variables that determine
    the cycle are to be copied across to the corresponding
    operational variables, expressed as a PTP timescale. The value
    of this variable is set by the SetConfigChangeTime() procedure
    in the List Config state machine.";
  reference
    "12.29.1.4 and 8.6.9.4.9 of IEEE Std 802.1Q-";
  uses ieee802:ptp-time-grouping;
}
leaf tick-granularity {
  type uint32;
  config false;
  description
    "Characteristics of an implementation's cycle timer clock

```

```
(TickGranularity).";
reference
  "8.6.9.4.16 of IEEE Std 802.1Q";
}
container current-time {
  config false;
  description
    "The current time maintained by the local system, expressed as a
    PTP timescale (see IEEE Std 802.1AS-2011).";
  reference
    "12.29.1.4 and 8.6.9.4.10 of IEEE Std 802.1Q-";
  uses ieee802:ptp-time-grouping;
}
leaf config-pending {
  type boolean;
  config false;
  description
    "A Boolean variable, set TRUE by the List Config state machine
    to signal that there is a new cycle configuration awaiting
    installation. The variable is set FALSE when the List Config
    state machine has installed the new configuration. The variable
    is used by the SetCycleStartTime() procedure to control the
    length of the cycle that immediately precedes the first cycle
    that uses the new configuration values. This value can be read
    by management.";
  reference
    "8.6.9.3 and 8.6.9.4.8 of IEEE Std 802.1Q-";
}
leaf config-change-error {
  type yang:counter64;
  config false;
  description
    "An error counter that is incremented if AdminBaseTime specifies
    a time in the past, and the current schedule is running.";
  reference
    "8.6.9.3.1 of IEEE Std 802.1Q";
}
leaf gate-closed-due-to-invalid-rx-enable {
  type boolean;
  default "false";
  description
    "A value of TRUE indicates that the GateClosedDueToInvalidRx
    function is enabled; a value of FALSE indicates that the
    GateClosedDueToInvalidRx function is disabled. The default value
    of GateClosedDueToInvalidRxEnable is FALSE.";
  reference
    "8.6.5.4 of IEEE Std 802.1Q";
}
leaf gate-closed-due-to-invalid-rx {
  type boolean;
  default "false";
  description
    "If GateClosedDueToInvalidRxEnable is TRUE, a value of TRUE in
    GateClosedDueToInvalidRx indicates that all frames are dropped
    (i.e., the gate behaves as if the operational stream gate state
    is Closed). If GateClosedDueToInvalidRx is FALSE, it has no
    effect. The default value of GateClosedDueToInvalidRx is FALSE;
    if any frame is discarded because the gate is in the Closed
    state, then GateClosedDueToInvalidRx is set TRUE.";
  reference
    "8.6.5.4 of IEEE Std 802.1Q";
}
leaf gate-closed-due-to-octets-exceeded-enable {
  type boolean;
  default "false";
  description
    "A value of TRUE indicates that the
    GateClosedDueToOctetsExceeded function is enabled; a value of
    FALSE indicates that the GateClosedDueToOctetsExceeded function
    is disabled. The default value of GateClosedDueToOctetsExceed is
    FALSE.";
  reference
```

```
        "8.6.5.4 of IEEE Std 802.1Q";
    }
    leaf gate-closed-due-octets-exceeded {
        type boolean;
        default "false";
        description
            "If GateClosedDueToOctetsExceededEnable is TRUE, a value of TRUE
            in GateClosedDueToOctetsExceeded indicates that all frames are
            dropped (i.e., the gate behaves as if the operational stream
            gate state is Closed). If GateClosedDueToOctetsExceeded is
            FALSE, it has no effect. The default value of
            GateClosedDueToOctetsExceeded is FALSE; if any frame is
            discarded because there are insufficient IntervalOctetsLeft,
            then GateClosedDueToOctetsExceeded is set TRUE.";
        reference
            "8.6.5.4 of IEEE Std 802.1Q";
    }
}
}
```

Insert 48.6.22 after 48.6.21 as follows:

48.6.22 The ieee802-dot1q-psfp-bridge YANG module

```
module ieee802-dot1q-psfp-bridge {
  yang-version "1.1";
  namespace urn:ieee:std:802.1Q:yang:ieee802-dot1q-psfp-bridge;
  prefix psfp-bridge;
  import ietf-ieee802-yang-dot1q-types-bridge {
    prefix yangdot1q;
  }
  import ieee802-types-dot1q-psfp {
    prefix ieee802psfp;
  }
import ieee802-dot1q-types {
  prefix dot1q-types;
}
import ieee802-dot1q-stream-filters-gates {
  prefix sfsg;
}
  organization
    "IEEE 802.1 Working Group";
  contact
    "WG-URL: http://www.ieee802.org/1/
    WG-EMail: stds-802-1-1@ieee.org

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

    E-mail: STDS-802-1-CHAIRS@IEEE.ORG";
  description
    "This module provides management of IEEE Std 802.1Q bridge components that Bridge
    components that support Per-Per-Stream Filtering and Policing (PSFP).

    Copyright (C) IEEE (2023).

    This version of this YANG module is part of IEEE Std 802.1Q; see the
    standard itself for full legal notices.";
  revision 2023-0310-17-26 {
    description
      "Published as part of IEEE Std 802.1Qcw-2023.

      The following reference statement identifies each referenced IEEE
      Standard as updated by applicable amendments.";
    reference
      "IEEE Std 802.1Q Bridges and Bridged Networks:
      IEEE Stds-Std 802.1Q-2022, IEEE Std 802.1Qcz-2023, IEEE Std 802.1Qcw-2023."*;
  }
  feature psfp {
    description
      "Per-Per-Stream Filtering and Policing supported.";
    reference
      "IEEE Std 802.1Q";
  }
identity set-gate-and-ipv {
  base dot1q-types:type-of-operation;
  description
    "The StreamGateState parameter specifies a desired state, open or
    closed, for the stream gate, and the IPV parameter specifies a desired
    value of the IPV associated with the stream. On execution, the
    StreamGateState and IPV parameter values are used to set the
    operational values of the stream gate state and internal priority
    specification parameters for the stream. After TimeInterval-ticks has
    elapsed since the completion of the previous stream gate control
    operation in the stream gate control list, control passes to the next
    stream gate control operation. The optional IntervalOctetMax parameter
    specifies the maximum number of MSDU octets that are permitted to pass
```

```

the gate during the specified TimeInterval. If the IntervalOctetMax
parameter is omitted, there is no limit on the number of octets that
can pass the gate."}
reference
"8.6.5.4 of IEEE Std 802.1Q"
}
grouping stream-gate-control-entries {
description
"A GateControlEntry consists of an operation name, followed by three
mandatory parameters and one optional parameter associated with the
operation. The first parameter is a gateStatesValue, the second
parameter is an IPV value, the third parameter is a TimeIntervalValue,
and the fourth parameter is an IntervalOctetMaxValue.
IntervalOctetMaxValue is optional."
uses dot1q-types:base-gate-control-entries {
refine "gate-control-entry/operation-name" {
must
"(.= 'psfp:set-gate-and-ipv!')"
}
refine "gate-control-entry/time-interval-value" {
must
"(.( <= ../../../supported-interval-max ))"
}
augment "gate-control-entry" {
description
"Augment gate-control-entry from base-gate-control-entries with
parameters gate-state-value, ipv-spec and interval-octet-max."
leaf gate-state-value {
type sfsg:gate-state-value-type;
mandatory true;
description
"The PSFPGateStatesValue indicates the desired gate state, open
or closed, for the stream gate."
reference
"12.31.3.2.1 of IEEE Std 802.1Q"
}
leaf ipv-spec {
type sfsg:ipv-spec-type;
mandatory true;
description
"The IPV value indicates the IPV to be associated with frames
that pass the gate."
reference
"12.31.3.2.3 of IEEE Std 802.1Q"
}
leaf interval-octet-max {
type uint32;
description
"An unsigned integer, denoting a IntervalOctetMax in MSDU
octets. If this parameter is omitted, there is no maximum."
reference
"12.31.3.2.5 of IEEE Std 802.1Qcw"
}
}
}
}
grouping psfp-parameters {
description
"psfp-parameters comprises all the parameters associated with
Per-Stream Filtering and Policing configuration."
container flow-meters {
description
"This container comprises all flow-meter related nodes."
list flow-meter-instance-table {
must
"(count(.) <= ../max-flow-meter-instances)" {
error-message
"Number of elements in flow-meter-instance-table must not be "+
"greater than max-flow-meter-instances."
}
key "flow-meter-instance-id"
description

```

```

    "There is one Flow Meter Instance Table per Bridge component. Each
    table row contains a set of parameters that defines a single Flow
    Meter Instance. 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 flow meters."
  reference
    "12.31.4 of IEEE Std 802.1Q";
  leaf flow-meter-instance-id {
    type uint32;
    mandatory true;
    description
      "An integer table index that allows the Flow Meter to be
      referenced from Stream Filter Instance Table entries."
    reference
      "8.6.5.5 of IEEE Std 802.1Q";
  }
  leaf committed-information-rate {
    type uint64;
    units "bits/second";
    mandatory true;
    description
      "Committed information rate (CIR), in bits per second."
    reference
      "8.6.5.5 of IEEE Std 802.1Q";
  }
  leaf committed-burst-size {
    type uint32;
    units "octets";
    mandatory true;
    description
      "Committed burst size (CBS), in octets."
    reference
      "8.6.5.5 of IEEE Std 802.1Q";
  }
  leaf excess-information-rate {
    type uint64;
    units "bits/second";
    mandatory true;
    description
      "Excess Information Rate (EIR), in bits per second."
    reference
      "8.6.5.5 of IEEE Std 802.1Q";
  }
  leaf excess-burst-size {
    type uint32;
    units "octets";
    mandatory true;
    description
      "Excess burst size (EBS) per bandwidth profile flow, in octets."
    reference
      "8.6.5.5 of IEEE Std 802.1Q";
  }
  leaf coupling-flag {
    type enumeration {
      enum zero {
        value 0;
        description
          "Uncoupled";
      }
      enum one {
        value 1;
        description
          "Coupled";
      }
    }
    mandatory true;
    description
      "Coupling flag (CF), which takes the value 0 or 1."
    reference
      "8.6.5.5 of IEEE Std 802.1Q";

```



```

    }
    leaf color-mode {
      type enumeration {
        enum color-blind {
          description
            "Color-blind (i.e., the color of a frame is ignored by the
            bandwidth profile algorithm).";
        }
        enum color-aware {
          description
            "Color-aware (i.e., the color of a frame is considered by
            the bandwidth profile algorithm).";
        }
      }
      mandatory true;
      description
        "Color mode (CM), which takes the value color-blind or
        color-aware, as specified in Bandwidth Profile Parameters and
        Algorithm in MEF 10.3.";
      reference
        "8.6.5.5 of IEEE Std 802.1Q";
    }
    leaf drop-on-yellow {
      type boolean;
      mandatory true;
      description
        "DropOnYellow, which takes the value TRUE or FALSE. A value of
        TRUE indicates that yellow frames are dropped (i.e., discarded);
        a value of FALSE indicates that yellow frames will have the
        drop_eligible parameter set to TRUE.";
      reference
        "8.6.5.5 of IEEE Std 802.1Q";
    }
    leaf mark-all-frames-red-enable {
      type boolean;
      default "false";
      description
        "MarkAllFramesRedEnable, which takes the value TRUE or FALSE. A
        value of TRUE indicates that the MarkAllFramesRed function is
        enabled; a value of FALSE indicates that the MarkAllFramesRed
        function is disabled. The default value of
        MarkAllFramesRedEnable is FALSE.";
      reference
        "8.6.5.5 of IEEE Std 802.1Q";
    }
    leaf mark-all-frames-red {
      type boolean;
      default "false";
      description
        "MarkAllFramesRed, which takes the value TRUE or FALSE. If
        MarkAllFramesRedEnable is TRUE, a value of TRUE in
        MarkAllFramesRed indicates that all frames are dropped (i.e.,
        discarded). If MarkAllFramesRed is False, it has no effect. The
        default value of MarkAllFramesRed is FALSE; if the operation of
        the flow meter causes any frame to be discarded, then
        MarkAllFramesRed is set TRUE.";
      reference
        "8.6.5.5 of IEEE Std 802.1Q";
    }
  }
}

leaf max-flow-meter-instances {
  type uint32;
  description
    "The maximum number of Flow Meter instances supported by this
    Bridge component. The object may optionally be read-only.";
  reference
    "12.31.1.3 of IEEE Std 802.1Q";
}
}

uses sfsg:sfsg-parameters {
  augment "stream-filters/stream-filter-instance-table" {
    description

```

```

    "Augment the Bridge component Stream filter by frame counters.";
    leaf matching-frames-count {
        type yang:counter64;
        config false;
        description
            "A count of frames matching both the stream_handle and priority
            specifications.";
        reference
            "8.6.5.3 of IEEE Std 802.1Q";
    }
    leaf passing-frames-count {
        type yang:counter64;
        config false;
        description
            "A count of frames that passed the stream gate.";
        reference
            "8.6.5.3 of IEEE Std 802.1Q";
    }
    leaf not-passing-frames-count {
        type yang:counter64;
        config false;
        description
            "A count of frames that did not pass the stream gate.";
        reference
            "8.6.5.3 of IEEE Std 802.1Q";
    }
    leaf red-frames-count {
        type yang:counter64;
        config false;
        description
            "A count of frames that were discarded as a result of the
            operation of the flow meter.";
        reference
            "8.6.5.3 of IEEE Std 802.1Q";
    }
    leaf passing-sdu-count {
        type yang:counter64;
        config false;
        description
            "A count of frames that passed the Maximum SDU size filter.";
        reference
            "8.6.5.3 of IEEE Std 802.1Q";
    }
    leaf not-passing-sdu-count {
        type yang:counter64;
        config false;
        description
            "A count of frames that did not pass the Maximum SDU size
            filter.";
        reference
            "8.6.5.3 of IEEE Std 802.1Q";
    }
    leaf flow-meter-ref {
        type leafref {
            path
                '.,'+
                '/..'+
                '/..'+
                '/flow-meters'+
                '/flow-meter-instance-table'+
                '/flow-meter-instance-id';
        }
        description
            "Flow meter instance that is associated with the stream filter.";
        reference
            "8.6.5.5 and 12.31.2.6 of IEEE Std 802.1Q.";
    }
    leaf flow-meter-enable {
        type boolean;
        description
            "If it is set to TRUE, flow-meter-instance-id identifies the
            flow meter instance associated with the stream filter. If it is

```

```

set to FALSE, no flow meter instance is associated with the
stream filter."}
reference
"8.6.5.5 and 12.31.2.6 of IEEE Std 802.1Q."
}
}
augment "stream-gates" {
description
"Augment the Bridge component Stream-Gates by maximum control list
limits, as used for PTP-controlled open and close transitions";
leaf-supported-list-max {
type uint32;
description
"The maximum value supported by this Bridge component of the
AdminControlListLength and OperControlListLength parameters. It
is available for use by schedule computation software to
determine the Bridge component's control list capacity prior to
computation. The object may optionally be read-only."}
reference
"12.31.1.4 of IEEE Std 802.1Q"
}
container-supported-cycle-max {
description
"The maximum value supported by this Bridge component of the
AdminCycleTime and OperCycleTime parameters. The object may
optionally be read-only."}
reference
"12.31.1.5 of IEEE Std 802.1Qcw"
uses ieee802:rational-grouping;
}
leaf-supported-interval-max {
type uint32;
description
"The maximum value supported by this Bridge component of the
TimeIntervalValue parameter. The object may optionally be
read-only."}
reference
"12.31.1.6 of IEEE Std 802.1Qcw"
}
}
augment "stream-gates/stream-gate-instance-table" {
description
"Augment the Bridge component Stream-Gate instances by
a) operational gate states
b) operational IPV values
c) PTP-controlled open and close transitions
d) Management for PTP-controlled open and close transitions";
leaf-oper-gate-state {
type sfsg:gate-state-value-type;
config false;
description
"The current operational state of the gate. The operational
state of the gate is set by the List-Execute state machine, and
its initial value is determined by the value of the
administrative state of the gate."}
reference
"12.31.3.2.1 and 8.6.10.5 of IEEE Std 802.1Q."
}
leaf-oper-ipv {
type sfsg:ipv-spec-type;
description
"The operational internal priority value specification."
reference
"12.31.3.3, 8.6.10.7 and 8.6.5.4 of IEEE Std 802.1Q."
}
container-admin-control-list {
must
"(count(/gate-control-entry) <= ../supported-list-max)" {
error-message
"Number of elements in admin-control-list must not be"
"greater than supported-list-max"
}
}

```

```

description
"The AdminControlList and OperControlList are ordered lists
containing AdminControlListLength or OperControlListLength
entries, respectively. Each entry represents a gate operation.
Each entry in the list is structured as a GateControlEntry."
reference
"12.31.3.2, 12.31.3.2.2 and 8.6.9.4.2 of IEEE Std 802.1Q."
uses stream-gate-control-entries;
}
container oper-control-list {
must
"(count(/gate-control-entry) <= ../../supported-list-max)" {
error-message
"Number of elements in oper-control-list must not be greater"
"than supported-list-max";
}
config false;
description
"The AdminControlList and OperControlList are ordered lists
containing AdminControlListLength or OperControlListLength
entries, respectively. Each entry represents a gate operation.
Each entry in the list is structured as a GateControlEntry."
reference
"12.31.3.2, 12.31.3.2.2 and 8.6.9.4.18 of IEEE Std 802.1Q."
uses stream-gate-control-entries;
}
container admin-cycle-time {
must
"(/numerator div ../denominator <= "+
"../../supported-cycle-max/numerator div "+
"../../supported-cycle-max/denominator)" {
error-message
"admin-cycle-time must not be greater than"
"supported-cycle-max";
}
description
"The administrative value of the gating cycle for the Port. This
value can be changed by management, and is used by the List
Config state machine to set the value of OperCycleTime. The
AdminCycleTime variable is a rational number of seconds, defined
by an integer numerator and an integer denominator."
reference
"12.29.1.3 and 8.6.9.4.3 of IEEE Std 802.1Q."
uses ieee802:rational-grouping;
}
container oper-cycle-time {
must
"(/numerator div ../denominator <= "+
"../../supported-cycle-max/numerator div "+
"../../supported-cycle-max/denominator)" {
error-message
"oper-cycle-time must not be greater than"
"supported-cycle-max";
}
config false;
description
"The operational value of the gating cycle for the Port. This
variable is set dynamically from the AdminCycleTime variable
under the control of the List Config state machine. OperCycleTime
is used by the Cycle Timer state machine to enforce the cycle
time for the Port. The OperCycleTime variable is a rational
number of seconds, defined by an integer numerator and an
integer denominator."
reference
"12.29.1.3 and 8.6.9.4.19 of IEEE Std 802.1Q."
uses ieee802:rational-grouping;
}
leaf admin-cycle-time-extension {
type uint32;
units "nanoseconds";
description
"An integer number of nanoseconds, defining the maximum amount

```

```

of time by which the gating cycle for the Port is permitted to
be extended when a new cycle configuration is being installed.
This administrative value can be changed by management, and is
used by the List-Config state machine to set the value of
OperCycleTimeExtension."
reference
"8.6.9.4.4 of IEEE Std 802.1Q"
}
leaf oper-cycle-time-extension {
type uint32
units "nanoseconds"
config false
description
"An integer number of nanoseconds, defining the maximum amount
of time by which the gating cycle for the Port is permitted to
be extended when a new cycle configuration is installed. This
operational value is set by the List-Config state machine to the
value of AdminCycleTimeExtension. The value of
OperCycleTimeExtension is used by the SetCycleStartTime()
augment "/dot1q:bridges/dot1q:bridge/dot1q:component" {
procedure."
description
reference
"8.6.9.4.20 of IEEE Std 802.1QAugment bridge with Per-Stream Filtering and
Policing configuration";
}
container admin-base-time {
description
"The administrative value of base-time, expressed as an IEEE
1588 precision time protocol (PTP) timescale (see IEEE Std
802.1AS-2011). This value can be changed by management, and is
used by the List-Config state machine to set the value of
OperBaseTime."
reference
"12.29.1.4 and 8.6.9.4.1 of IEEE Std 802.1Q."
uses ieee802:ptp-time-grouping
}
container oper-base-time {
config false
description
"The operational value of base-time, expressed as a PTP
timescale (see IEEE Std 802.1AS-2011). This variable is used by
the List-Config state machine."
uses psfp:psfp-parameters
reference
"12.29.1.4 and 8.6.9.4.17 of IEEE Std 802.1Q."
uses ieee802:ptp-time-grouping
}
leaf config-change {
type boolean
description
"A Boolean variable that acts as a start signal to the List
Config state machine that the administrative variable values for
the Port are ready to be copied into their corresponding
operational variables. This variable is set TRUE by management
and is set FALSE by the List-Config state machine."
reference
"8.6.9.4.7 of IEEE Std 802.1Q"
}
container config-change-time {
config false
description
"The time at which the administrative variables that determine
the cycle are to be copied across to the corresponding
operational variables, expressed as a PTP timescale. The value
of this variable is set by the SetConfigChangeTime() procedure
in the List-Config state machine."
reference
"12.29.1.4 and 8.6.9.4.9 of IEEE Std 802.1Q."
uses ieee802:ptp-time-grouping
}
leaf tick-granularity {
type uint32
config false

```

```

    description
    "Characteristics of an implementation's cycle timer clock
    (TickGranularity).";
    reference
    "8.6.9.4.16 of IEEE Std 802.1Q";
  }
  container current-time {
    config false;
    description
    "The current time maintained by the local system, expressed as a
    PTP timescale (see IEEE Std 802.1AS-2011).";
    reference
    "12.29.1.4 and 8.6.9.4.10 of IEEE Std 802.1Q";
    uses ieee802:ptp-time-grouping;
  }
  leaf-config pending {
    type boolean;
    config false;
    description
    "A Boolean variable, set TRUE by the List Config state machine
    to signal that there is a new cycle configuration awaiting
    installation. The variable is set FALSE when the List Config
    state machine has installed the new configuration. The variable
    is used by the SetCycleStartTime() procedure to control the
    length of the cycle that immediately precedes the first cycle
    that uses the new configuration values. This value can be read
    by management.";
    reference
    "8.6.9.3 and 8.6.9.4.8 of IEEE Std 802.1Q";
  }
  leaf-config change-error {
    type yang:counter64;
    config false;
    description
    "An error counter that is incremented if AdminBaseTime specifies
    a time in the past, and the current schedule is running.";
    reference
    "8.6.9.3.1 of IEEE Std 802.1Q";
  }
  leaf-gate-closed-due-to-invalid-rx-enable {
    type boolean;
    default "false";
    description
    "A value of TRUE indicates that the GateClosedDueToInvalidRx
    function is enabled; a value of FALSE indicates that the
    GateClosedDueToInvalidRx function is disabled. The default value
    of GateClosedDueToInvalidRxEnable is FALSE.";
    reference
    "8.6.5.4 of IEEE Std 802.1Q";
  }
  leaf-gate-closed-due-to-invalid-rx {
    type boolean;
    default "false";
    description
    "If GateClosedDueToInvalidRxEnable is TRUE, a value of TRUE in
    GateClosedDueToInvalidRx indicates that all frames are dropped
    (i.e., the gate behaves as if the operational stream gate state
    is Closed). If GateClosedDueToInvalidRx is FALSE, it has no
    effect. The default value of GateClosedDueToInvalidRx is FALSE;
    if any frame is discarded because the gate is in the Closed
    state, then GateClosedDueToInvalidRx is set TRUE.";
    reference
    "8.6.5.4 of IEEE Std 802.1Q";
  }
  leaf-gate-closed-due-octets-exceeded-enable {
    type boolean;
    default "false";
    description
    "A value of TRUE indicates that the
    GateClosedDueToOctetsExceeded function is enabled; a value of
    FALSE indicates that the GateClosedDueToOctetsExceeded function
    is disabled. The default value of GateClosedDueToOctetsExceed is

```

```

_____ FALSE,"}
_____ reference
_____ "8.6.5.4 of IEEE Std 802.1Q"}
_____ }
_____ leaf gate-closed-due-octets-exceeded {
_____ type boolean}
_____ default "false"}
_____ description
_____ "If GateClosedDueToOctetsExceededEnable is TRUE, a value of TRUE
_____ in GateClosedDueToOctetsExceeded indicates that all frames are
_____ dropped (i.e., the gate behaves as if the operational stream
_____ gate state is Closed). If GateClosedDueToOctetsExceeded is
_____ FALSE, it has no effect. The default value of
_____ GateClosedDueToOctetsExceeded is FALSE; if any frame is
_____ discarded because there are insufficient IntervalOctetsLeft,
_____ then GateClosedDueToOctetsExceeded is set TRUE."}
_____ reference
_____ "8.6.5.4 of IEEE Std 802.1Q"}
_____ }
_____ }
_____ }
}

```