

IEEE P802.1CBcv

Usage of *AutonomousType* in the MIB



Stephan Kehrer, Hirschmann Automation and Control GmbH

September 2020

Motivation

- Several comments on the MIBs presented in draft D0.4 of IEEE P802.1CBcv asked for clarification on the usage of *AutonomousType* in the MIB
- This presentation is intended to provide some hints and clarifications on the construct of *AutonomousType* and its usage

What is *AutonomousType*?

- *AutonomousType* is a textual convention defined in [RFC 2579](#)
- The description in the RFC is:
Represents an independently extensible type identification value. It may, for example, indicate a particular sub-tree with further MIB definitions, or define a particular type of protocol or hardware.
- It allows to use sub-tables (i.e. tables that are conceptually part of another table) by providing a means to reference them by “pointer”

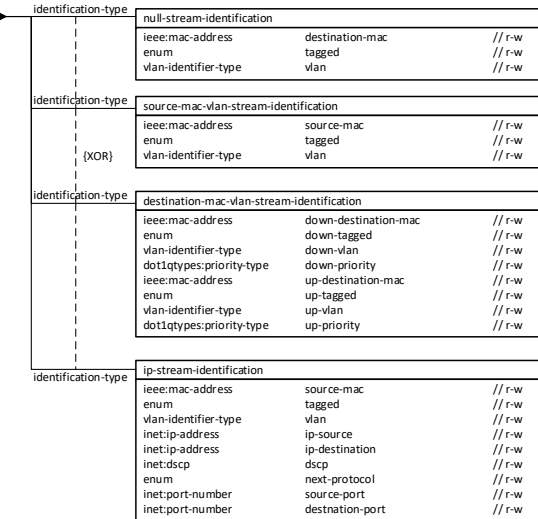
Usage in IEEE P802.3CBcv

- IEEE Std 802.3CB contains several tables that
 - require flexibility in the amount of entries they can contain
 - require flexibility in the managed objects that are part of the sub-tree**but** are included in another set of managed objects that are modeled as a table in the MIB
- Examples are:
 - the stream identity table, clause 9.1
 - the sequence generation table, clause 10.3
 - ...

Example *tsnStreamIdentificationType* (clause 9.1.1.6)

stream-identity		
uint32	index;	// not accessible
uint32	handle	// r
if-name	* in-facing-input-port-list	// r-w
if-name	* in-facing-output-port-list	// r-w
if-name	* out-facing-input-port-list	// r-w
if-name	* out-facing-output-port-list	// r-w
enum	identification-type	// r
parameters		

- The *tsnStreamIdentification* type defines the stream identification function used to identify a specific stream
- Depending on the stream identification function a dedicated subset of parameters is required (i.e. Null Stream identification requires different objects than IP Stream identification)
- This is modeled in the MIB using *AutonomousType* as shown on the next slides
- The usage of *AutonomousType* to model the stream identification type is modeled the same way as *hrDeviceType* is modeled in [RFC 2790](#)



Example *tsnStreamIdentificationType* (clause 9.1.1.6)

- Definition of the *StreamIdentificationTypes*
- Definition of an object *StreamIdentificationTypeSelect* in the stream identification table
 - is of type *AutonomousType*
 - contains one of the previously defined *StreamIdentificationTypes*
 - description clarifies which sub-table is used for the respective identification types
- Definition of the sub-tables with the required set of managed objects for the respective stream identification methods
- Sub-table entry is linked to the parent table by using the index of the parent entry as its own index

Example *tsnStreamIdentificationType* (clause 9.1.1.6)

- Definition of the *StreamIdentificationTypes*

```
-- the ieee8021StreamIdStreamIdentification table
-- =====
-- the ieee8021StreamIdStreamIdentificationTypes
-- =====
ieee8021StreamIdStreamIdentificationTypes
  OBJECT IDENTIFIER ::= { ieee8021StreamIdStreamIdentification 1 }

ieee8021StreamIdNullStream OBJECT-IDENTITY
  STATUS current
  DESCRIPTION
    "The stream identification type for the Null Stream
    identification."
  ::= { ieee8021StreamIdStreamIdentificationTypes 1 }

ieee8021StreamIdSrcMacVlan OBJECT-IDENTITY
  STATUS current
  DESCRIPTION
    "The stream identification type for the Source MAC and VLAN
    Stream identification."
  ::= { ieee8021StreamIdStreamIdentificationTypes 2 }

ieee8021StreamIdActiveDestMacVlan OBJECT-IDENTITY
  STATUS current
  DESCRIPTION
    "The stream identification type for the Active Destination
    and VLAN Stream identification."
  ::= { ieee8021StreamIdStreamIdentificationTypes 3 }

ieee8021StreamIdIpStream OBJECT-IDENTITY
  STATUS current
  DESCRIPTION
    "The stream identification type for the IP Stream
    identification."
  ::= { ieee8021StreamIdStreamIdentificationTypes 4 }
```

- Definition of an object *StreamIdentificationTypeSelect* in the stream identification table

```
ieee8021StreamIdStreamIdIdentificationTypeSelect OBJECT-TYPE
  SYNTAX AutonomousType
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "An indication of the type of stream identification method that
    is used.

    If this value is
    'ieee8021StreamIdNullStream ieee8021StreamIdObjects 1' then
    an entry exists in the
    ieee8021StreamIdStreamIdentificationTable
    which corresponds to the Null Stream identification.

    If this value is
    'ieee8021StreamIdSrcMacVlan ieee8021StreamIdObjects 2' then
    an entry exists in the
    ieee8021StreamIdStreamIdentificationTable
    which corresponds to the Source MAC and VLAN Stream
    identification.

    If this value is
    'ieee8021StreamIdActiveDestMacVlan ieee8021StreamIdObjects 3'
    then an entry exists in the
    ieee8021StreamIdStreamIdentificationTable
    which corresponds to the Active Destination MAC and VLAN Stream
    identification.

    If this value is
    'ieee8021StreamIdIpStream ieee8021StreamIdObjects 4' then
    an entry exists in the
    ieee8021StreamIdStreamIdentificationTable
    which corresponds to the IP Stream identification."
  ::= { ieee8021StreamIdStreamIdentificationEntry 6 }
```

Example *tsnStreamIdentificationType* (clause 9.1.1.6)

- Definition of the sub-tables with the required set of managed objects for that stream identification
- Sub-table entry is linked to the parent table by using the index of the parent entry as its own index

```
-- =====  
-- the ieee8021StreamIdNullStreamIdentificationTable  
-- =====  
  
ieee8021StreamIdNullStreamIdentificationTable OBJECT-TYPE  
SYNTAX SEQUENCE OF Ieee8021StreamIdNullStreamIdentificationEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "A table containing a set of controlling parameters for the  
    Null Stream identification method."  
REFERENCE "9.1.2"  
 ::= { ieee8021StreamIdStreamIdentification 3 }  
  
ieee8021StreamIdNullStreamIdentificationEntry OBJECT-TYPE  
SYNTAX Ieee8021StreamIdNullStreamIdentificationEntry  
MAX-ACCESS not-accessible  
STATUS current  
DESCRIPTION  
    "A set of managed objects that serve as the Stream  
    identification parameters when using the Null Stream  
    identification method."  
REFERENCE "9.1.2"  
 INDEX { ieee8021StreamIdStreamIdentificationIndex }  
 ::= { ieee8021StreamIdNullStreamIdentificationTable 1 }  
  
Ieee8021StreamIdNullStreamIdentificationEntry ::=  
SEQUENCE {  
    ieee8021StreamIdCpeNullDownDestMac  
        MacAddress,  
    ieee8021StreamIdCPENullDownTagged  
        Ieee8021CBTaggedType,  
    ieee8021StreamIdCpeNullDownVlan  
        Ieee8021CBVlanIdentifier  
}  
  
ieee8021StreamIdCpeNullDownDestMac OBJECT-TYPE  
SYNTAX MacAddress  
MAX-ACCESS read-write  
STATUS current
```


Example *tsnStreamIdInFacOutputPortList* (clause 9.1.1.2)

- *tsnStreamIdInFacOutputPortList* contains a list of ports “[...] on which an in-facing Stream identification function using this identification method is to be placed for this Stream in the output (towards the system forwarding function) direction[...]”
 - This is modeled in the MIB using *AutonomousType* as shown on the next slides
 - The length of the list is unknown, so other ways to model this (e.g. OCTET STRING) did not seem appropriate or unnecessarily restraining
 - The 802.1 bridge MIB defines a type *PortList*. Can’t we use that instead?
 - *PortList* is effectively a bitmap representing the physical ports existing in a bridge
 - In IEEE Std 802.1CB the usage of port does not refer to physical bridge ports, it refers to interfaces
- For this reason we can not use *PortList*; the entries in the port lists modeled in the MIB need to be references to interfaces, not to physical bridge ports

Example *tsnStreamIdInFacOutputPortList* (clause 9.1.1.2)

- Definition of an object *ieee8021StreamIdStreamIdInFacOutputPortList* in the stream identification table
 - is of type *AutonomousType*
 - contains OID of the sub-table that actually holds the port/interface entries
 - description clarifies which sub-table is used for the respective identification types
- Definition of the sub-table with the required set of managed objects for the port list
- Sub-table entry is linked to the parent table by using the index of the parent entry as its own index

Example *tsnStreamIdInFacOutputPortList* (clause 9.1.1.2)

- Definition of an object
StreamIdentificationTypeSelect in
the stream identification table

```
ieee8021StreamIdStreamIdInFacOutputPortList OBJECT-TYPE
  SYNTAX      AutonomousType
  MAX-ACCESS  read-create
  STATUS      current
  DESCRIPTION
    "The list of ports on which an in-facing Stream identification
     function using this identification method is to be placed for
     this Stream in the output (towards the system forwarding
     function) direction."
  REFERENCE   "9.1.1.2"
  ::= { ieee8021StreamIdStreamIdentificationEntry 8 }
```

Example *tsnStreamIdInFacOutputPortList* (clause 9.1.1.2)

- Definition of the sub-table with the required set of managed objects for the port list
- Sub-table entry is linked to the parent table by using the index of the parent entry as its own index
 - additional index is required because each entry in the sub-table contains exactly one interface
 - a list of interfaces in an entry of the parent table requires several entries in the sub-table

```
-- =====
-- the ieee8021StreamIdStreamIdInFacOutputPortHandleList
-- table
-- =====
ieee8021StreamIdStreamIdInFacOutputPortHandleListTable OBJECT-TYPE
SYNTAX      SEQUENCE OF Ieee8021StreamIdStreamIdInFacOutputPortHandleListEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A table containing a list of ports on which an in-facing Stream
    identification function using this identification method is to
    be placed for this Stream in the output (towards the system
    forwarding function) direction, referenced in
    ieee8021StreamIdStreamIdentificationIndex."
REFERENCE   "9.1.1.2"
 ::= { ieee8021StreamIdStreamIdInFacOutputPortHandleList 2 }

ieee8021StreamIdStreamIdInFacOutputPortHandleListEntry OBJECT-TYPE
SYNTAX      Ieee8021StreamIdStreamIdInFacOutputPortHandleListEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A set of managed objects providing the ports on which an
    in-facing Stream identification function using this
    identification method is to be placed for this Stream in the
    output (towards the system forwarding function) direction,
    referenced in ieee8021StreamIdStreamIdentificationIndex."
REFERENCE   "9.1.1.2"
INDEX       { ieee8021StreamIdStreamIdentificationIndex,
              ieee8021StreamIdStreamIdInFacOutputPortHandleListIndex
            }
 ::= { ieee8021StreamIdStreamIdInFacOutputPortHandleListTable 1 }

Ieee8021StreamIdStreamIdInFacOutputPortHandleListEntry ::=
SEQUENCE {
    ieee8021StreamIdStreamIdInFacOutputPortHandleListIndex
        Unsigned32,
    ieee8021StreamIdStreamIdInFacOutputPortHandle
        VariablePointer,
    ieee8021StreamIdStreamIdInFacOutputPortHandleListStatus
        RowStatus
}

ieee8021StreamIdStreamIdInFacOutputPortHandleListIndex OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  not-accessible
```

Thank you!

Any questions?