Ethernet Pseudo Wire (PW) Management Information Base

draft-ietf-pwe3-enet-mib-08.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt
The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects for modeling of Ethernet Pseudo Wire (PW) services.
1 Introduction

This document describes a model for managing Ethernet pseudo wire services for transmission over a packet Switched Network (PSN). This MIB module is generic and common to all types of PSNs supported in the PWE3 architecture [RFC4446], which describes the transport and encapsulation of L1 and L2 services over supported PSN types.

In particular, the MIB module associates a port or specific VLANs on top of a physical Ethernet port or a virtual Ethernet interface (for VPLS service) to a point-to-point PW. It is complementary to the [PWMIB], which is used to manage the generic PW parameters common to all service, including all supported PSN types.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [BCP14].
2 Terminology

This document uses terminology from the document describing the PW architecture [RFC3985] and from [RFC4448].

3 The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

4 Feature Checklist

The PW Ethernet MIB module (PW-ENET-STD-MIB) is designed to satisfy the following requirements and constraints:

- The MIB module is designed to be work with the PW-STD-MIB [PWMIB].
- The MIB module is independent of the PSN type.
- The MIB module supports various options for selecting Ethernet packets into the PW. These include port-based PW, VLAN-based PW, VLAN-change and adding or removing VLAN fields between the port to be emulated and the PW.
- In the case of an MPLS PSN, the MIB module supports the use of multiple PWs to carry the same Ethernet service. These PWs can be used to support L-LSPs or single COS E-LSPs capable PSN, when mapping of the Ethernet PRI bits to the PSN COS is required.
- The MIB module enables both point-to-point Ethernet services and VPLS services as discussed in the L2VPN working group [VPLS].
- The MIB module allows modeling of the PW as an Ethernet virtual port to be managed via existing Ethernet MIBs like Etherlike-MIB [RFC3635].
5 PW-MIB usage

The MIB module structure for defining a PW service is composed of three layers of MIB modules functioning together. This general model is defined in the PWE3 architecture [RFC3985]. The layering model is intended to sufficiently isolate PW services from the underlying PSN layer that carries the emulated service. This is done at the same time as providing a standard means for connecting any supported services to any supported PSNs.

The first layer known as the service layer contains service-specific modules such as the one defined in this document. These modules define service-specific management objects that interface or collaborate with existing MIB modules for the native version of the service. The service-specific module "glues" the standard modules to the PWE3 MIB modules.

The next layer of the PWE3 MIB framework is the PW MIB module [PWMIB]. This module is used to configure general parameters of PWs that are common to all types of emulated services and PSNs. This layer is connected to the service-specific layer above, and the PSN layer below.

The PSN layer provides PSN-specific modules for each type of PSN. These modules associate the PW with one or more "tunnels" that carry the service over the PSN. These modules are defined in other documents. This module is used to "glue" the PW service to the underlying PSN-specific MIB modules.

[PWTC] defines some of the object types used in these modules.

The Etherlike-MIB [RFC3635] does not support virtual Ethernet port, however it is sometimes desired to manage the PW as an Ethernet port via the Etherlike-MIB. This MIB module supports an option to recognize the PW as an ifIndex, enabling standard use of the Etherlike-MIB to manage the PW.

5.1 PW-ENET-MIB usage

- The PW table (pwTable) is used for all PW types (ATM, FR, Ethernet, SONET, etc.). This table contains high level generic parameters related to the PW creation. A row is typically created by the operator (see [PWMIB] for other options) for each PW service.

- Based on the PSN type defined for the PW, rows are created in PSN specific module (for example [PWMPLSMIB]) and associated to the pwTable by the common pwIndex.
- If the PW type is Ethernet or EthernetTagged a row is created by the agent in pwEnetTable.

- When using a MPLS PSN, it may be required to separate the same Ethernet services to multiple PW in order to support multiple COS on the same service. In this case, multiple PWs, each with the appropriate COS will be created to the same destination, and classification will be based also on the Ethernet PRI bits marking. The MIB allow any combinations of multiple PRI setting to PSN COS mapping (The exact PSN marking (EXP bits, DSCP etc.) is out of scope of this MIB module). In these cases, pwEnetTable will hold multiple rows with the same Ethernet port and VLAN mapping, each PW will need to be created separately by the signaling process.

5.2 PW-ENET management model

The management model for the Ethernet PW is shown in figure 1, and is based on the PW layering [RFC3985].

```
+--------------------------------------+
|                PE Device             |
+--------------------------------------+
     Single |                 |                    |
     AC     |                 |      Single        | PW Instance
<------>o   Forwarder     +      PW Instance   X<========>
     Single |                 |                    |
+--------------------------------------+
```

May be modeled as ifIndex

Notation:
- o       A physical CE-bound PE port
- +       A PW IWF instance interface to the forwarder.
- X       A PE PSN-bound port.

Figure 1: A simple point-to-point service

In the typical point-to-point service, the object pwEnetPortIfIndex associates the physical CE-bound PE port (‘o’) to the PW (it is allowed to have multiple PWs associated to the same physical port). This MIB module also manages some of the possible operations of the forwarder.

In some models it is convenient to model the forwarder virtual interface to a PW IWF instance (‘+’) as an ifIndex. As discussed in
the [PWMIB], this is possible by using the PW ifType in the ifTable and indicating the ifIndex in the main pwTable. In case of Ethernet PW a virtual interface of ifType = etherLike will be assigned on top of the PW interface to enable statistics gathering and statuses and other management configuration tasks via existing tools. This way, the PW instance is managed as virtual Ethernet interface in the PE.

The model for using the PW in non-point to point applications, such as VPLS are done with the same principle in mind, except that the creation of the tables is related typically to an auto-discovery process.

5.3 Example of MIB usage

Assume we would like to create a PW of type VLAN between two PEs, for VLAN value 5.

- Follows the example in [PWMIB], except that the pwType is equal ‘ethernetTagged’.

- The agent creates a row in pwEnetTable and pwEnetStatsTable for the specified pwIndex. The pwEnetPwInstance is automatically selected by the agent to the value of ‘1’.

- The operator fills the following entries in the pwEnetTable:

  pwEnetPwVlan        5,
  pwEnetVlanMode      noChange,
  pwEnetPortVlan      5,
  pwEnetPortIfIndex   1001,
  pwEnetPwIfIndex     0, -- Not managed in the
                      -- Etherlike MIB module

- The PW is ready for forwarding when signaling has been accomplished successfully between the two peers.

5.4 Service delimiting modes

In this section we will describe how the MIB module supports point-to-point applications with various VLAN service delimiting options on the physical port and the PW modes and VLAN values. In the VPLS case the PW is attached to a virtual interface that is attached to a bridge or VPLS forwarder, and it is assumed that this bridge function would be responsible for the packet modifications between the physical port or other virtual port and the virtual port the PW is attached to.
There are 3 main services that are supported by this MIB module.

1) Port mode: In this mode, the whole traffic from the port is mapped to the PW.

1a) In the typical application, the packet is sent to the PW as is:

        pwEnetPwVlan        4097,
        pwEnetVlanMode      portMode,
        pwEnetPortVlan      4097,

        pwType              Ethernet,

1b) It is possible to add a provider tag (value 10 for example) to the packet when it is sent over the PW:

        pwEnetPwVlan        10,
        pwEnetVlanMode      addVlan,
        pwEnetPortVlan      4097,

‘EthernetTagged’ SHOULD be used as the pwType.

2) Single VLAN: In this mode, only the first VLAN field on the packet from the physical port is the service delimiting tag, as an example VLAN=5. The following options of processing are possible:

2a) One to one mapping: the service delimiting tag is kept as is on the PW.

        pwEnetPwVlan        5,
        pwEnetVlanMode      noChange,
        pwEnetPortVlan      5,

‘EthernetTagged’ SHOULD be used as the pwType.

2b) One to one mapping: the service delimiting tag change it’s value (to value of 6) on the PW.

        pwEnetPwVlan        6,
        pwEnetVlanMode      changeVlan,
        pwEnetPortVlan      5,

‘EthernetTagged’ SHOULD be used as the pwType.

2c) The service delimiting tag is removed when the packet is sent to the PW.

        pwEnetPwVlan        5,
        pwEnetVlanMode      removeVlan,
        pwEnetPortVlan      4097,
'EthernetTagged' SHOULD be used as the pwType.

It should note that this mode is also applicable when the service delimiting tag is a service provider tag (VLAN=5 in this case), and the service provider actually replace the map the traffic to a single PW independent of the packet format on top of this VLAN.

2d) Untagged packets mapped to a PW as is (packets with a VLAN field from the same port MAY be mapped to other PWs).

\[
\begin{align*}
pwEnetPwVlan & \quad 4096, \\
pwEnetVlanMode & \quad noChange, \\
pwEnetPortVlan & \quad 4096,
\end{align*}
\]

pwType MAY equal ‘Ethernet’ or ‘EthernetTagged’.

2e) Untagged packets mapped to a PW, and a VLAN field is added to the packet.

\[
\begin{align*}
pwEnetPwVlan & \quad 6, \\
pwEnetVlanMode & \quad addVlan, \\
pwEnetPortVlan & \quad 4096,
\end{align*}
\]

‘EthernetTagged’ SHOULD be used as the pwType.

2f) A provider VLAN (value 10) is added to the packet coming with VLAN value 5 before it is sent to the PW.

\[
\begin{align*}
pwEnetPwVlan & \quad 10, \\
pwEnetVlanMode & \quad addVlan, \\
pwEnetPortVlan & \quad 5,
\end{align*}
\]

‘EthernetTagged’ SHOULD be used as the pwType.

3) Nested VLAN (QinQ): When only the first VLAN is the service delimiting tag, the modes as described in 2) SHOULD be used. In other cases, i.e. the service delimiting tag is both the first VLAN and the second VLAN, the following option is supported by this MIB module. Other scenarios are considered out of the scope and should be handled by other MIB modules that manage the forwarder.

Assuming the provider tag equals 5 and the (inner) VLAN equal 100, this traffic can be mapped to the PW without the provider tag by using the following configuration:

\[
\begin{align*}
pwEnetPwVlan & \quad 100, \\
pwEnetVlanMode & \quad removeVLAN, \\
pwEnetPortVlan & \quad 5,
\end{align*}
\]
pwType MAY equal 'Ethernet' or 'EthernetTagged', 'EthernetTagged' is RECOMMENDED.

Packets of the same provider tag MAY be mapped to other PWs.

6 Object definitions

PW-ENET-STD-MIB DEFINITIONS ::= BEGIN

IMPORTS
OBJECT-TYPE, MODULE-IDENTITY, Unsigned32, transmission
FROM SNMPv2-SMI -- [RFC2578]

MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF -- [RFC2580]

StorageType, RowStatus
FROM SNMPv2-TC -- [RFC2579]

InterfaceIndexOrZero
FROM IF-MIB -- [RFC2863]

ZeroBasedCounter32
FROM RMON2-MIB -- [RFC2021]

pwIndex
FROM PW-STD-MIB -- [PWMIB]
-- RFC Editor: Please replace
-- PWMIB with correct RFC #

PwVlanCfg
FROM PW-TC-STD-MIB; -- [PWTC]
-- RFC Editor: Please replace
-- PWTC with correct RFC #

pwEnetStdMIB MODULE-IDENTITY
LAST-UPDATED "200606141200Z" -- 14 June 2006 12:00:00 GMT
ORGANIZATION "IETF PWE3 Working group"
CONTACT-INFO
"David Zelig
E-mail: davidz@corrigent.com

Thomas D. Nadeau
Email: tnadeau@cisco.com"

DESCRIPTION
"This MIB module describes a model for managing Ethernet point-to-point pseudo wire services over a Packet Switched Network (PSN)."
Copyright (C) The Internet Society (2006). This version of this MIB module is part of RFC yyyy; see the RFC itself for full legal notices.

```
-- RFC Editor: replace yyyy with actual RFC number & remove
-- this note

-- Revision history.
REVISION "200606141200Z" -- 14 June 2006 12:00:00 GMT
DESCRIPTION "Initial version published as part of RFC XXXX."
-- RFC Editor: replace XXXX with actual RFC number & remove
-- this note

::= { transmission XXXX } -- RFC Editor: please replace
   -- XXXX with IANA assigne value.
   -- See IANA considerations sect.

pwEnetObjects OBJECT IDENTIFIER ::= { pwEnetStdMIB 1 }
pwEnetConformance OBJECT IDENTIFIER ::= { pwEnetStdMIB 2 }

--
-- Ethernet PW table
--

pwEnetTable OBJECT-TYPE
SYNTAX SEQUENCE OF PwEnetEntry
MAX-ACCESS  not-accessible
STATUS     current
DESCRIPTION
   "This table contains the index to the Ethernet tables associated with this ETH PW, the VLAN configuration and VLAN mode."
::= { pwEnetObjects 1 }

pwEnetEntry OBJECT-TYPE
SYNTAX     PwEnetEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
   "This table is indexed by the same index that was created for the associated entry in the PW generic table in the PW-STD-MIB module. The pwIndex and the pwEnetPwInstance are used as indexes to allow multiple VLANs to exist on the same PW.

An entry is created in this table by the agent for every entry in the pwTable with a pwType of ‘ethernetTagged’ or ‘ethernet’. Additional rows may be created by the operator or the agent if multiple entries are required for
the same PW.

The value of pwEnetPwInstance can be arbitrary selected to make the row unique, however implementations that know the VLAN field value when the row is created MAY use the value of the VLAN itself for better readability and backward compatibility with older versions of this MIB module.

This table provides Ethernet port mapping and VLAN configuration for each Ethernet PW.

INDEX { pwIndex, pwEnetPwInstance }
::= { pwEnetTable 1 }

PwEnetEntry ::= SEQUENCE {
  pwEnetPwInstance    Unsigned32,
  pwEnetPwVlan        PwVlanCfg,
  pwEnetVlanMode      INTEGER,
  pwEnetPortVlan      PwVlanCfg,
  pwEnetPortIfIndex   InterfaceIndexOrZero,
  pwEnetPwIfIndex     InterfaceIndexOrZero,
  pwEnetRowStatus     RowStatus,
  pwEnetStorageType   StorageType
}

pwEnetPwInstance OBJECT-TYPE
SYNTAX      Unsigned32
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"If multiple rows are mapped to the same PW, this index is used to uniquely identify the individual row. If the value of the VLAN field is known at the time of row creation, the value of pwEnetPwVlan MAY be used for better readability and backward compatibility with older versions of this MIB module. Otherwise the value of ‘1’ SHOULD be set to the first row for each pwIndex for better readability and in order that the management application would know in advance how to access the first row when it was created by the agent."

 ::= { pwEnetEntry 1 }

pwEnetPwVlan OBJECT-TYPE
SYNTAX      PwVlanCfg
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION

"This Object defines the (service delimiting) VLAN field value on the PW. The value of 4097 MUST be used if the object is not applicable, for example when mapping all packets from an Ethernet port to this PW (raw mode). The value of 4096 MUST be set to indicate untagged frames (from the PW point of view), i.e. when pwEnetVlanMode equals ‘noChange’ and pwEnetPortVlan equals 4096."

::= { pwEnetEntry 2 }

pwEnetVlanMode  OBJECT-TYPE
SYNTAX     INTEGER {
    other(0),
    portBased(1),
    noChange(2),
    changeVlan(3),
    addVlan(4),
    removeVlan(5)
}
MAX-ACCESS  read-create
STATUS     current
DESCRIPTION

"Indicates the mode of VLAN handling between the port or the virtual port associated to the PW and the PW encapsulation.

- ‘other’ indicates an operation that is not defined by this MIB module.
- ‘portBased’ indicates that the forwarder will forward packets between the port and the PW independent of their structure (i.e. there is no service delimiting VLAN tags from the PE standpoint).
- ‘noChange’ indicates that the PW contains the original user VLAN, as specified in pwEnetPortVlan, i.e. the VLAN on the PE-CE link is the service delimiting tag and is kept as is on the PW.
- ‘changeVlan’ indicates that the VLAN field on the PW may be different than the VLAN field on the user’s port. the VLAN on the PE-CE link is the service delimiting tag but has a different value on the PW.
- ‘removeVlan’ indicates that the encapsulation on the PW does not include the service delimiting VLAN field. Note that PRI bits transparency is lost in this case.
- ‘addVlan’ indicate that a VLAN field will be added on the PSN bound direction. pwEnetPwVlan indicates the value that will be added."
Implementation of 'portsbased', 'removeVlan', 'addVlan', 'other' and 'changeVlan' is OPTIONAL.

DEFVAL { noChange }
 ::= { pwEnetEntry 3 }

pwEnetPortVlan OBJECT-TYPE
SYNTAX     PwVlanCfg
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
 "This object defines the VLAN value on the physical port (or
VPLS virtual port) or a mapping of the whole port traffic
to the same PW.

The value of '4097' MUST be used if the whole port traffic
is mapped to the same PW. Note that a pwType of
'ethernetTagged' can still be used if service delimiting tag
is added on the PW (pwEnetVlanMode equals 'addVlan').

It MUST be equal to pwEnetPwVlan if pwEnetVlanMode
equals 'noChange'.

The value 4096 indicates that packet without a VLAN field
(i.e. untagged frames) on the port are associated to this
PW. This allows the same behaviors as assigning 'Default
VLAN' to untagged frames.

DEFVAL { 4097 }
 ::= { pwEnetEntry 4 }

pwEnetPortIfIndex  OBJECT-TYPE
SYNTAX      InterfaceIndexOrZero
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
 "This object is used to specify the ifIndex of the Ethernet
port associated with this PW for point-to-point Ethernet
service, or the ifIndex of the virtual interface of the
VPLS instance associated with the PW if the service is
VPLS. Two rows in this table can point to the same ifIndex
only if:

1) It is required to support multiple COS on a MPLS PSN
for the same service (i.e.: a combination of ports and
VLANS) by the use of multiple PW, each with a different
COS.

2) There is no overlap of VLAN values specified in
pwEnetPortVlan that are associated with this port.
A value of zero indicates that association to an ifIndex is not yet known.

::= { pwEnetEntry 5 }

pwEnetPwIfIndex OBJECT-TYPE
SYNTAX InterfaceIndexOrZero
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"If the PW is modeled as an ifIndex in the ifTable, this
object indicates the value of the ifIndex representing the
Ethernet PW on the PSN side in the Etherlike-MIB. Note that
this value may be different from the value of pwIfIndex
that represent the ifIndex of the PW for ifType 'pw'."
DEFVAL { 0 }
::= { pwEnetEntry 6 }

pwEnetRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION
" Enables creating, deleting and modifying this row."
::= { pwEnetEntry 7 }

pwEnetStorageType OBJECT-TYPE
SYNTAX StorageType
MAX-ACCESS read-create
STATUS current
DESCRIPTION
"Indicates the storage type of this row."
::= { pwEnetEntry 8 }

--
-- Ethernet PW Statistics Table
--

pwEnetStatsTable OBJECT-TYPE
SYNTAX SEQUENCE OF PwEnetStatsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table contains statistical counters specific for
Ethernet PW."
::= { pwenetObjects 2 }

PwEnetStatsEntry OBJECT-TYPE
SYNTAX PwEnetStatsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"Each entry represents the statistics gathered for the PW carrying the Ethernet packets since this PW was first created in the pwEnetTable."
INDEX { pwIndex }
::= { pwEnetStatsTable 1 }

PwEnetStatsEntry ::= SEQUENCE {
pwEnetStatsIllegalVlan ZeroBasedCounter32,
pwEnetStatsIllegalLength ZeroBasedCounter32
}

pwEnetStatsIllegalVlan OBJECT-TYPE
SYNTAX ZeroBasedCounter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of packets received (from the PSN) on this PW with an illegal VLAN field, missing VLAN field that was expected, or a VLAN field when it was not expected. This counter may not be applicable in some case, and MUST return the value of zero in this case."
 ::= { pwEnetStatsEntry 1 }

pwEnetStatsIllegalLength OBJECT-TYPE
SYNTAX ZeroBasedCounter32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The number of packets that were received with an illegal Ethernet packet length on this PW. An illegal length is defined as being greater than the value in the advertised MTU supported, or shorter than the allowed Ethernet packet size. The agent should start the value of this counter at the value of zero."
 ::= { pwEnetStatsEntry 2 }

---
--- Conformance description
---

pwEnetGroups OBJECT IDENTIFIER ::= { pwEnetConformance 1 }
pwEnetCompliances OBJECT IDENTIFIER ::= { pwEnetConformance 2 }

-- Compliance requirement for fully compliant implementations.
pwEnetModuleFullCompliance MODULE-COMPLIANCE
STATUS current
DESCRIPTION
"The compliance statement for agents that provide full
support for PW-ENET MIB Module. Such devices can
then be monitored and also be configured using
this MIB module."

MODULE -- this module
MANDATORY-GROUPS {
    pwEnetGroup,
    pwEnetStatsGroup
}

OBJECT       pwEnetVlanMode
DESCRIPTION "An implementation MUST support at least the value
noChange(2)."

OBJECT       pwEnetPwIfIndex
MIN-ACCESS   read-only
DESCRIPTION "Write access and values other than zero are
required only for implementations that support the
modeling the Ethernet PW in the Etherlike-MIB."

OBJECT       pwEnetRowStatus
SYNTAX       RowStatus { active(1), notInService(2),
             notReady(3) }
WRITE-SYNTAX RowStatus { active(1), notInService(2),
             createAndGo(4), destroy(6) }
MIN-ACCESS   read-only
DESCRIPTION "Support for createAndWait is not required. Support
of notReady is not required for implementations that
do not support signaling.
Support of read-write is not required for
implementations that do not support more than one
VLAN mapping to the same PW."

::= { pwEnetCompliances 1 }

-- Compliance requirement for read-only compliant implementations.

pwEnetModuleReadOnlyCompliance MODULE-COMPLIANCE
STATUS  current
DESCRIPTION "The compliance statement for agents that provide read
only support for PW-ENET MIB Module. Such devices can
then be monitored but cannot be configured using this
MIB module."

MODULE -- this module
MANDATORY-GROUPS { pwEnetGroup,
                    pwEnetStatsGroup

Zelig et al          Expires December 2006               [page 16]
OBJECT pwEnetPwVlan
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

OBJECT pwEnetVlanMode
MIN-ACCESS read-only
DESCRIPTION "Write access is not required. An implementation
MUST support at least the value noChange(2)."

OBJECT pwEnetPortVlan
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

OBJECT pwEnetPortIfIndex
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

OBJECT pwEnetPwIfIndex
MIN-ACCESS read-only
DESCRIPTION "Write access is not required. Values other than
zero are required only for implementations that
support the modeling the Ethernet PW in the Etherlike-MIB."

OBJECT pwEnetRowStatus
SYNTAX RowStatus { active(1), notInService(2),
notReady(3) }  
MIN-ACCESS read-only
DESCRIPTION "Write access is not required. Support
of notReady is not required for implementations that
do not support signaling."

OBJECT pwEnetStorageType
MIN-ACCESS read-only
DESCRIPTION "Write access is not required."

::= { pwEnetCompliances 2 }

-- Units of conformance.

pwEnetGroup OBJECT-GROUP
OBJECTS {
  pwEnetPwVlan,
  pwEnetVlanMode,
  pwEnetPortVlan,
  pwEnetPortIfIndex,
  pwEnetPwIfIndex,
  pwEnetRowStatus,
  pwEnetStorageType}
7 Security Considerations

It is clear that this MIB module is potentially useful for monitoring of Ethernet PW capable PEs. This MIB module can also be used for configuration of certain objects, and anything that can be configured can be incorrectly configured, with potentially disastrous results.

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations. These are the tables and objects and their sensitivity/vulnerability:

- the pwEnetTable contains objects to provision Ethernet PWs. Unauthorized access to objects in these tables, could result in disruption of traffic on the network. The use of stronger mechanisms such as SNMPv3 security should be considered where possible. Specifically, SNMPv3 VACM and USM MUST be used with any v3 agent which implements this MIB module.

  Administrators should consider whether read access to these objects should be allowed, since read access may be undesirable under certain circumstances.

Some of the readable objects in this MIB module "i.e., objects with a MAX-ACCESS other than not-accessible" may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly
to even encrypt the values of these objects when sending them over
the network via SNMP. These are the tables and objects and their
sensitivity/vulnerability:

- the pwEnetTable shows the Ethernet PW service configuration.
  If an Administrator does not want to reveal this
  information, then these tables should be
  considered sensitive/vulnerable.

SNMP versions prior to SNMPv3 did not include adequate security.
Even if the network itself is secure "for example by using IPSec",
even then, there is no control as to who on the secure network is
allowed to access and GET/SET "read/change/create/delete" the
objects in this MIB module.

It is RECOMMENDED that implementers consider the security features
as provided by the SNMPv3 framework "see [RFC3410], section 8",
including full support for the SNMPv3 cryptographic mechanisms "for
authentication and privacy".

Further, deployment of SNMP versions prior to SNMPv3 is NOT
RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to
enable cryptographic security. It is then a customer/operator
responsibility to ensure that the SNMP entity giving access to an
instance of this MIB module, is properly configured to give access
to the objects only to those principals "users" that have legitimate
rights to indeed GET or SET "change/create/delete" them.

8 IANA considerations

The MIB module in this document uses the following IANA-assigned
OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>OBJECT IDENTIFIER value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pwEnetStdMIB</td>
<td>{ transmission XXXX }</td>
</tr>
</tbody>
</table>

Editor’s Note (to be removed prior to publication): the IANA is
requested to assign a value for "XXXX" under the ‘transmission’
subtree and to record the assignment in the SMI Numbers registry.
When the assignment has been made, the RFC Editor is asked to
replace "XXXX" (here and in the MIB module) with the assigned value
and to remove this note.

9 References

9.1 Normative references


Ethernet Pseudo Wire (PW)             June 2006
Management Information Base


9.2 Informative references


10 Author’s Addresses

David Zelig
Corrigent Systems
126, Yigal Alon st.
Tel Aviv, ISRAEL
Phone: +972-3-6945273
Email: davidz@corrigent.com

Thomas D. Nadeau
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Email: tnadeau@cisco.com
11 Full Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

12 Intellectual Property Notice

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.