

Network Working Group
Request for Comments: 5603
Category: Standards Track

D. Zelig, Ed.
Oversight
T. Nadeau, Ed.
BT
July 2009

Ethernet Pseudowire (PW) Management Information Base (MIB)

Abstract

This memo defines a 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 pseudowire (PW) services.

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	2
2. The Internet-Standard Management Framework	2
3. Conventions	3
4. Overview	3
5. Feature Checklist	4
6. PW ENET MIB Module Usage	4
7. PW-ENET Management Model	5
8. Example of the PW-ENET MIB Module Usage	6
9. Service-Delimiting Modes	6
10. Object Definitions	9
11. Security Considerations	19
12. IANA Considerations	21
13. References	21
13.1. Normative References	21
13.2. Informative References	22
14. Acknowledgments	22

1. Introduction

This document describes a model for managing Ethernet pseudowire services for transmission over a Packet Switched Network (PSN). This MIB module is generic and common to all types of PSNs supported in the Pseudowire Emulation Edge-to-Edge (PWE3) architecture [RFC3985], 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 Virtual Private LAN Service (VPLS)) to a point-to-point PW. It is complementary to the PW-STD-MIB [RFC5601], which manages the generic PW parameters common to all services, including all supported PSN types.

2. 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 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].

3. Conventions

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].

This document adopts the definitions, acronyms, and mechanisms described in [RFC3985] and [RFC3916]. Unless otherwise stated, the mechanisms of [RFC3985] apply and will not be re-described here.

4. Overview

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. 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 [RFC5601]. 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 used to "glue" the PW service to the underlying PSN-specific MIB modules. This document defines the MIB module for Ethernet PW over any PSN type.

This module uses Textual Conventions (TCs) and objects as defined in [RFC2578], [RFC2579], [RFC2580], [RFC2863], [RFC4363], [RFC4502], and [RFC5601].

The Etherlike-MIB [RFC3635] does not support virtual Ethernet ports; 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. Feature Checklist

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

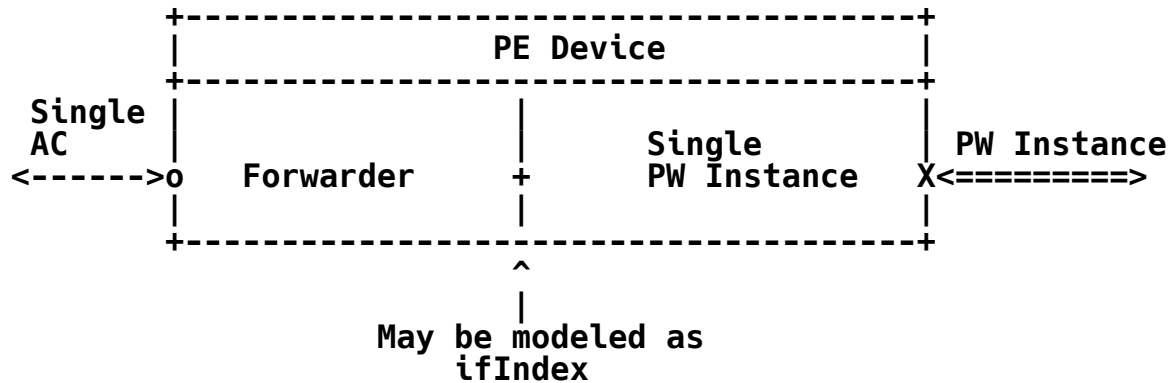
- The MIB module is designed to work with the PW-STD-MIB [RFC5601].
- The MIB module is agnostic to the PSN type.
- The MIB module supports various options for selecting Ethernet packets into the PW, as defined in [RFC4448]. These include port-based PW, VLAN-based PW, and VLAN-manipulated based (change, add, or remove) 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 Label-Only-Inferred-PSC LSPs (L-LSPs) or EXP-Inferred-PSC LSPs (E-LSPs) that are from a single Class of Service (CoS), when mapping of the Ethernet user priority (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 [RFC4664].
- The MIB module allows modeling of the PW as an Ethernet virtual port to be managed via existing Ethernet MIBs like Etherlike-MIB [RFC3635].

6. PW ENET MIB Module 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 [RFC5542] for other options) for each PW service.
- Based on the PSN type defined for the PW, rows are created in the PSN-specific module (for example, [RFC5602]) 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 the pwEnetTable.

7. PW-ENET Management Model

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



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 [RFC5601], 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, is done with the same principle in mind, except that the creation of the tables is related typically to an auto-discovery process.

8. Example of the PW-ENET MIB Module Usage

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

- Follows the example in [RFC5601], with pwType equals 'ethernetTagged'.
- The agent creates a row in the pwEnetTable and a row in the pwEnetStatsTable for the specified pwIndex. The pwEnetPwInstance is automatically set 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.

9. Service-Delimiting Modes

This section describes how the MIB module supports point-to-point applications with various VLAN service-delimiting options on the original Ethernet port and the corresponding PW mode and VLAN values. If the PW is attached to VPLS service, the PW is associated to a virtual interface that is attached to a bridge or VPLS forwarder. The bridging function between local physical ports and virtual interfaces that are later associated to PWs is not handled via this MIB module.

There are three main service types that are supported by this MIB module:

- (1) Port mode: In this mode, the whole traffic from the port is mapped to the PW.
 - A. In the typical application, the packet is sent to the PW as is:

pwEnetPwVlan	4095,
pwEnetVlanMode	portMode,
pwEnetPortVlan	4095,
pwType	Ethernet,
 - B. 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	4095,

pwType SHOULD be set to 'EthernetTagged'.
- (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:
 - A. One-to-one mapping: The service-delimiting tag is kept as is on the PW.

pwEnetPwVlan	5,
pwEnetVlanMode	noChange,
pwEnetPortVlan	5,

pwType SHOULD be set to 'EthernetTagged'.
 - B. VLAN change mapping: The service-delimiting tag changes its value (to the value of 6) on the PW.

pwEnetPwVlan	6,
pwEnetVlanMode	changeVlan,
pwEnetPortVlan	5,

pwType SHOULD be set to 'EthernetTagged'.

- C. The service-delimiting tag is removed when the packet is sent to the PW.

```
pwEnetPwVlan      4095,  
pwEnetVlanMode    removeVlan,  
pwEnetPortVlan    5,
```

pwType SHOULD be set to 'EthernetTagged'.

It should be noted that this mode is also applicable when the service-delimiting tag is a service provider tag (VLAN=5 in this case), and the node removes this VLAN and maps the traffic to a single PW independent of the packet format on top of this VLAN.

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

```
pwEnetPwVlan      0,  
pwEnetVlanMode    noChange,  
pwEnetPortVlan    0,
```

pwType MAY equal 'Ethernet' or 'EthernetTagged'.

- E. Untagged packets mapped to a PW, and a VLAN field is added to the packet.

```
pwEnetPwVlan      6,  
pwEnetVlanMode    addVlan,  
pwEnetPortVlan    0,
```

pwType SHOULD be set to 'EthernetTagged'.

- F. A provider VLAN (value 10) is added to packets arriving with VLAN value 5 before they are sent to the PW.

```
pwEnetPwVlan      10,  
pwEnetVlanMode    addVlan,  
pwEnetPortVlan    5,
```

pwType SHOULD be set to 'EthernetTagged'.

- (3) Nested VLAN (QinQ): When only the first VLAN is the service-delimiting tag, one of the modes as described in 2) SHOULD be used. If the service-delimiting tag is both the first VLAN and the second VLAN, the following option is supported by this MIB module:

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

```
pwEnetPwVlan      100,
pwEnetVlanMode    removeVLAN,
pwEnetPortVlan    5,
```

It is RECOMMENDED that the pwType would equal 'EthernetTagged', but pwType equal to 'Ethernet' MAY be used as well.

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

- (4) Other scenarios are considered out of scope and should be handled by other MIB modules that manage the forwarder and the Native Service Processing (NSP) sections.

10. Object Definitions

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

IMPORTS

OBJECT-TYPE, MODULE-IDENTITY, Unsigned32, mib-2
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 -- [RFC4502]

pwIndex
FROM PW-STD-MIB -- [RFC5601]

VlanIdOrAnyOrNone
FROM Q-BRIDGE-MIB; -- [RFC4363]

pwEnetStdMIB MODULE-IDENTITY

LAST-UPDATED "200906150000Z" -- 15 June 2009 00:00:00 GMT

ORGANIZATION "Pseudowire Edge-to-Edge Emulation (PWE3) Working Group"

CONTACT-INFO

"David Zelig
Email: davidz@oversi.com

Thomas D. Nadeau
Email: tom.nadeau@bt.com
"

DESCRIPTION

"This MIB module describes a model for managing Ethernet point-to-point pseudowire services over a Packet Switched Network (PSN).

Copyright (c) 2009 IETF Trust and the persons identified as authors of the code. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Internet Society, IETF or IETF Trust, nor the names of specific contributors, may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This version of this MIB module is part of RFC 5603;

see the RFC itself for full legal notices."

-- Revision history

REVISION "200906150000Z" -- 15 June 2009 00:00:00 GMT

DESCRIPTION "Initial version published as part of RFC 5603."

::= { mib-2 180 }

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 Ethernet PW, the VLAN configuration, and the 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 arbitrarily 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.

All read-create objects in this table MAY be changed at any time; however, change of some objects (for example, pwEnetVlanMode) during the PW forwarding state MAY cause traffic disruption.

Manual entries in this table SHOULD be preserved after a reboot, and the agent MUST ensure the integrity of those entries. If the set of entries of a specific row is found to be inconsistent after reboot, the PW pwOperStatus MUST be declared as notPresent(5).

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

```
PwEnetEntry ::= SEQUENCE {
    pwEnetPwInstance      Unsigned32,
    pwEnetPwVlan          VlanIdOrAnyOrNone,
    pwEnetVlanMode        INTEGER,
    pwEnetPortVlan        VlanIdOrAnyOrNone,

    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
1 SHOULD be set to the first row for each pwIndex
for better readability and in order that the management
application will know in advance how to access the
first row when it was created by the agent."
```

```

"
 ::= { pwEnetEntry 1 }

pwEnetPwVlan OBJECT-TYPE
    SYNTAX      VlanIdOrAnyOrNone
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object defines the (service-delimiting) VLAN field
        value on the PW. The value 4095 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 0 MUST be set to indicate untagged frames
        (from the PW point of view), i.e., when pwEnetVlanMode
        equals 'noChange' and pwEnetPortVlan equals 0."
 ::= { 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
        "This object indicates the mode of VLAN handling between the
        port or the virtual port associated with 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 are 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.

- 'addVlan' indicates that a VLAN field will be added on the PSN-bound direction (i.e., on the PW). pwEnetPwVlan indicates the value that will be added.
- '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.
- Implementation of 'portsbased', 'removeVlan', 'addVlan', 'other', and 'changeVlan' is OPTIONAL.

```
DEFVAL { noChange }
::= { pwEnetEntry 3 }
```

```
pwEnetPortVlan OBJECT-TYPE
SYNTAX      VlanIdOrAnyOrNone
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION
```

"This object defines if the mapping between the original port (physical port or VPLS virtual port) to the PW is VLAN based or not. In case of VLAN mapping, this object indicates the VLAN value on the original port.

The value of '4095' MUST be used if the whole original 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').

This object MUST be equal to pwEnetPwVlan if pwEnetVlanMode equals 'noChange'.

The value 0 indicates that packets without a VLAN field (i.e., untagged frames) on the port are associated to this PW. This allows the same behavior as assigning 'Default VLAN' to untagged frames.

```
DEFVAL { 4095 }
::= { 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 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 represents the ifIndex of the PW for ifType 'pw'."

```
DEFVAL { 0 }
```

```
::= { pwEnetEntry 6 }
```

```
pwEnetRowStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"This object enables creating, deleting, and modifying this row."

```
::= { pwEnetEntry 7 }
```

```
pwEnetStorageType OBJECT-TYPE
```

```
SYNTAX      StorageType
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"This object indicates the storage type of this row."

```
DEFVAL { nonVolatile }
```

```
::= { 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."
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 either an illegal VLAN field, a missing VLAN field
 when one was expected, or an excessive VLAN field when
 it was not expected. This counter may not be applicable
 in some cases, and MUST return the value of zero in
 such a 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."


```

 ::= { 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 provides full
        support for the PW-ENET-STD-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 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 the PW-ENET-STD-MIB module. Such devices can then be monitored but cannot be configured using this MIB module."

MODULE -- this module

MANDATORY-GROUPS { pwEnetGroup,
pwEnetStatsGroup
}

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 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

}
 STATUS current
 DESCRIPTION

"Collection of objects for basic Ethernet PW configuration."

::= { pwEnetGroups 1 }

pwEnetStatsGroup OBJECT-GROUP
 OBJECTS {

pwEnetStatsIllegalVlan,
 pwEnetStatsIllegalLength

}
 STATUS current
 DESCRIPTION

"Collection of objects counting various PW level errors."

::= { pwEnetGroups 2 }

END

11. Security Considerations

It is clear that this MIB module is potentially useful for monitoring of 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:

- o 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 that 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:

- o 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.

12. IANA Considerations

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

Descriptor -----	OBJECT IDENTIFIER value -----
pwEnetStdMIB	{ mib-2 180 }

13. References

13.1. Normative References

- [BCP14] Bradner, S., "Key words for use in RFCs to Indicate requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3635] Flick, J., "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 3635, September 2003.
- [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, April 2006.
- [RFC4502] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2", RFC 4502, May 2006.
- [RFC4363] Levi, D. and D. Harrington, "Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering, and Virtual LAN Extensions", RFC 4363, January 2006.

- [RFC5601] Zelig, D., Ed., and T. Nadeau, Ed., "Pseudowire (PW) Management Information Base (MIB)", RFC 5601, July 2009.

13.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.
- [RFC3916] Xiao, X., Ed., McPherson, D., Ed., and P. Pate, Ed., "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, September 2004.
- [RFC3985] Bryant, S., Ed., and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.
- [RFC4664] Andersson, L., Ed., and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, September 2006.
- [RFC5542] Nadeau, T., Ed., Zelig, D., Ed., and O. Nicklass, Ed., "Definitions of Textual Conventions for Pseudowires (PW) Management", RFC 5542, May 2009.
- [RFC5602] Zelig, D., Ed., and T. Nadeau, Ed., "Pseudowire (PW) over MPLS PSN Management Information Base (MIB)", RFC 5602, July 2009.

14. Acknowledgments

This document was produced by the PWE3 Working Group. Special thanks to Orly Nicklass for close review and good suggestions.

Authors' Addresses

David Zelig (editor)
Oversi Networks
1 Rishon Letzion St.
Petah Tikva
Israel

Phone: +972 77 3337 750
EMail: davidz@oversi.com

Thomas D. Nadeau (editor)
BT
BT Centre
81 Newgate Street
London EC1A 7AJ
United Kingdom

EMail: tom.nadeau@bt.com