Ethernet Pseudo Wire (PW) Management Information Base
draft-pwe3-enet-mib-04.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

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.

Copyright (C) The Internet Society (2001). All rights reserved.

1 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.
Table of Contents

1 Abstract.......................................................1
2 Introduction...................................................2
3 Terminology...................................................3
4 The Internet-Standard Management Framework.....................3
5 Feature Checklist.............................................3
6 PW-MIB usage..................................................3
6.1 PW-ENET-MIB usage..........................................4
6.2 PW-ENET management model................................5
6.3 Example of MIB usage........................................6
7 Object definitions...............................................6
8 Security considerations........................................17
9 IANA considerations...........................................18
10 References.....................................................19
10.1 Normative references........................................19
10.2 Informative references.....................................20
11 Author’s Addresses...........................................20
12 Full Copyright Statement......................................20

2 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 [FRARCH], 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].
3 Terminology

This document uses terminology from the document describing the PW architecture [FRARCH] and from [PW-ENET].

4 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].

5 Feature Checklist

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

- The MIB module is designed to be work with the PW-MIB [PW-MIB].
- 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].

6 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 [FRARCH]. 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 ÂgluesÂ the standard modules to the PWE3 MIB modules.

The next layer of the PWE3 MIB framework is comprised of 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 ÂglueÂ the PW service to the underlying PSN-specific MIB modules. In the case of MPLS, for example, the PW MPLS MIB [PWMPLSMIB] is used to connect the PW service to either the MPLS-LDP [LDPMIB] or MPLS-TE [TEMIB] MIBs. [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 support an option to recognize the PW as an ifIndex, enabling standard use of the Etherlike-MIB to manage the PW.

6.1 PW-ENET-MIB usage

- The PW table (pwVcTable) 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 pwVcTable by the common pwVcIndex.
- If the PW type is Ethernet or EthernetTagged a row is created by the agent in pwVcEnetTable.

- 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). In these cases, pwVcEnetTable will hold multiple rows with the same Ethernet port and VLAN mapping, each PW will need to be created separately by the signaling process.

6.2 PW-ENET management model

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

```
+--------------------------------------+
<table>
<thead>
<tr>
<th>PE Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
</tr>
</tbody>
</table>
|<------>o   Forwarder     +      PW Instance X<========>
|                 |                    |
+--------------------------------------+
```

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 pwVcEnetPortIfIndex 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 pwVcTable. 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.

6.3 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 pwVcType is equal EthernetTagged.

- The agent creates a row in pwVcEnetTable and pwVcEnetStatsTable for the specified pwVcindex.

- The operator fills the following entries in the pwVcEnetTable:

  pwVcEnetPwVlan 5,
pwVcEnetVlanMode noChange,
pwVcEnetPortVlan 5,
pwVcEnetPortIfIndex 1001,
pwVcEnetVcIfIndex 0, -- Not managed in the
   -- Etherlike MIB module
   ...

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

7 Object definitions

-- Ethernet PW MIB
--

PW-ENET-DRAFT04-MIB DEFINITIONS ::= BEGIN

IMPORTS
   OBJECT-TYPE, MODULE-IDENTITY
FROM SNMPv2-SMI

Zelig et al Expires June 2003 [page 6]
MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF

StorageType, RowStatus
FROM SNMPv2-TC

InterfaceIndexOrZero
FROM IF-MIB

ZeroBasedCounter32
FROM RMON2-MIB

pwVcIndex
FROM PW-DRAFT04-MIB

pwStdMIB, PwVcVlanCfg
FROM PW-TC-DRAFT04-MIB;

cpyvMIB MODULE-IDENTITY
LAST-UPDATED "200402041200Z" -- 4 February 2004 12:00:00 GMT
ORGANIZATION "IETF PWE3 Working group"
CONTACT-INFO
"David Zelig
Postal: Corrigent Systems
126, Yigal Alon St.
Tel-Aviv, ISRAEL
Tel: +972-3-6945273
E-mail: davidz@corrigent.com

Thomas D. Nadeau
Postal: Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Tel:    +1-978-497-3051
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)."

-- Revision history.

REVISION
"200402041200Z" -- 4 February 2004 12:00:00 GMT
DESCRIPTION
" Changes from as draft-pwe3-enet-mib-03:
  1) Remove pwVcEnetFragSize and pwVcEnetCwStatus since they are controlled by the PW MIB module and they are duplicated here.
  2) Change counter64 to ZeroBasedCounter32 in error\n\nZelig et al Expires June 2003
statistics, and fix them to be read-only.

REVISION
"200312161200Z" -- 16 December 2003 12:00:00 GMT
DESCRIPTION
" Changes from as draft-pwe3-enet-mib-02.
  1) Module identity under the PW MIB tree.
"

REVISION
"200307291200Z" -- 29 July 2003 12:00:00 GMT
DESCRIPTION
" Changes from as draft-pwe3-enet-mib-01.
  1) Added Control Word status and fragmentation status.
"

REVISION
"200209221200Z" -- 22 September 2002 12:00:00 GMT
DESCRIPTION
" Submitted as draft-pwe3-enet-mib-00.
  Changes from previous version:
  1) Alignment with draft-pwe3-ethernet-encap-00.txt:
      removing 'rangeVLAN' mode and the associated objects.
  2) Relaxing requirement on value of pwVcEnetPortVlan in
      port mode.
"

REVISION
"200208201200Z" -- 20 August 2002 12:00:00 GMT
DESCRIPTION
"Changes from previous version:
  1) Add pwVcEnetVcIfIndex - Option for VC as ifIndex.
  2) Change counters to 64 bits.
  3) Add mode for adding/removing VLAN fields between PW and
     CE bound interface.
  4) Referencing draft-martini instead of draft-so.
  5) Editorial changes for some description clauses.
  6) MPLS PRI mapping table to be independent (not augmented).
  7) Adapt descriptions and rules of use to
draft-ietf-pwe3-Ethernet-encap-00.
"

REVISION
"200202031200Z" -- 03 February 2002 12:00:00 GMT
DESCRIPTION
"initial revision as -00 draft"

::= { pwStdMIB 5 } -- To be assigned by IANA

pwVcEnetObjects OBJECT IDENTIFIER ::= { pwVcEnetMIB 1 }

Zelig et al Expires June 2003 [page 8]
Ethernet Pseudo Wire (PW)  
Management Information Base 

pwVcEnetConformance  OBJECT IDENTIFIER ::= { pwVcEnetMIB 2 }

--  Ethernet PW table  --

pwVcEnetTable  OBJECT-TYPE
SYNTAX  SEQUENCE OF PwVcEnetEntry
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."
 ::= { pwVcEnetObjects 1 }

pwVcEnetEntry  OBJECT-TYPE
SYNTAX  PwVcEnetEntry
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 Table in the PW MIB. The PwVcIndex and the pwVcEnetPwVlan 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 pwVcTable with a pwVcType of 'ethernetTagged' or 'ethernet'. Additional rows may be created by the operator or the agent if multiple entries are required for the same PW.

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

INDEX { pwVcIndex, pwVcEnetPwVlan }
 ::= { pwVcEnetTable 1 }

PwVcEnetEntry ::= SEQUENCE {}
 pwVcEnetPwVlan   PwVcVlanCfg,
pwVcEnetVlanMode  INTEGER,
pwVcEnetPortVlan  PwVcVlanCfg,
pwVcEnetPortIfIndex  InterfaceIndexOrZero,
pwVcEnetVcIfIndex  InterfaceIndexOrZero,
pwVcEnetRowStatus  RowStatus,
pwVcEnetStorageType  StorageType
 }

pwVcEnetPwVlan  OBJECT-TYPE
Ethernet Pseudo Wire (PW)         February 2004
Management Information Base

SYNTAX      PwVcVlanCfg
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
"This Object defines the VLAN on the PW. The value of 4097
is used if the object is not applicable, for example when
mapping all packets from an Ethernet port to this PW.
The value of 4096 is used to indicate untagged frames (from
the PW point of view), for example if pwVcEnetVlanMode
equals ‘removeVLAN’ or when pwVcEnetVlanMode equals
‘noChange’ and pwVcEnetPortVlan equals 4096."

::= { pwVcEnetEntry 1 }

pwVcEnetVlanMode  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
associated to the PW and the PW encapsulation itself.

- ‘other’ indicate 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.

- ‘noChange’ indicates that the PW contains the original
  user VLAN, as specified in pwVcEnetPortVlan.

- ‘changeVlan’ indicates that the VLAN field on the PW
  may be different than the VLAN field on the user’s
  port.

- ‘removeVlan’ indicates that the encapsulation on the
  PW does not include the original 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. pwVcEnetPwVlan indicate
  the value that will be added."
- 'removeVlan', 'addVlan' and 'changeVlan' implementation is not required.

DEFVAL { noChange }
::= { pwVcEnetEntry 2 }

pwVcEnetPortVlan OBJECT-TYPE
SYNTAX PwVcVlanCfg
MAX-ACCESS read-create
STATUS current
DESCRIPTION "This object define the VLAN value on the physical port (or VPLS virtual port) if a change is required to the VLAN value between the PW and the physical/virtual port.

The value of this object can be ignored if the whole traffic from the port is forwarded to one PW independent of the tagging on the port, but it is RECOMMENDED that the value in this case will be '4097' indicating not relevant.

It MUST be equal to pwVcEnetPwVlan if 'noChange' mode is used.

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 un-tagged frames."

DEFVAL { 4097 }
::= { pwVcEnetEntry 3 }

pwVcEnetPortIfIndex 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

Zelig et al Expires June 2003 [page 11]
pwVcEnetPortVlan that are associated with this port.

A value of zero indicate that association to an ifIndex is not yet known."

::= { pwVcEnetEntry 4 }

pwVcEnetVcIfIndex  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 pwVcIfIndex that represent the ifIndex of the PW for ifType ‘pw’.’"

DEFVAL { 0 }
::= { pwVcEnetEntry 5 }

pwVcEnetRowStatus  OBJECT-TYPE
SYNTAX     RowStatus
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"Enable creating, deleting and modifying this row."

::= { pwVcEnetEntry 6 }

pwVcEnetStorageType  OBJECT-TYPE
SYNTAX     StorageType
MAX-ACCESS read-create
STATUS     current
DESCRIPTION
"Indicates the storage type of this row."

::= { pwVcEnetEntry 7 }

--
-- Ethernet Priority Mapping Table
--

pwVcEnetMplsPriMappingTable  OBJECT-TYPE
SYNTAX SEQUENCE OF PwVcEnetMplsPriMappingTableEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION
"This table may be used for MPLS PSNs if there is a need to hold multiple PW, each with different COS, for the same
user service (port + PW VLAN). Such a need may arise if the MPLS network is capable of L-LSP or E-LSP without multiple COS capabilities. Each row is indexed by the pwVcIndex and indicate the PRI bits on the packet received from the user port (or VPLS virtual port) that are classified to this PW. Note that the EXP bit value of the PW is configured in the PW MPLS MIB module.

 ::= { pwVcEnetObjects 2 }

pwVcEnetMplsPriMappingTableEntry OBJECT-TYPE
SYNTAX     PwVcEnetMplsPriMappingTableEntry
MAX-ACCESS  not-accessible
STATUS     current
DESCRIPTION
"Each entry is created by the operator or by the agent based on local policy if special classification based on the PRI bits is required for this PW."

INDEX { pwVcIndex }

 ::= { pwVcEnetMplsPriMappingTable 1 }

PwVcEnetMplsPriMappingTableEntry ::= SEQUENCE {
  pwVcEnetMplsPriMapping             BITS,
  pwVcEnetMplsPriMappingRowStatus    RowStatus,
  pwVcEnetMplsPriMappingStorageType  StorageType
}

pwVcEnetMplsPriMapping  OBJECT-TYPE
SYNTAX     BITS {
  pri000 (0),
  pri001 (1),
  pri010 (2),
  pri011 (3),
  pri100 (4),
  pri101 (5),
  pri110 (6),
  pri111 (7),
  untagged (8)
}
MAX-ACCESS  read-create
STATUS     current
DESCRIPTION
"This object defines the groups of user PRI mapped into this PW. Each bit set indicates that this user priority is assigned to this PW.

The value ‘untagged’ is used to indicate that untagged frames are also associated to this PW."
This object allow the use of different PSN COS based on user marking of PRI bits in MPLS PSN with L-LSP or E-LSP without multiple COS support. In all other cases, the default value MUST be used.

It is REQUIRED that there is no overlap on this object between rows serving the same service (port+ PW VLAN).

In case of missing BIT configuration between rows to the same service, incoming packets with PRI marking not configured should be handled by the PW with the lowest COS.

REFERENCE

"See appendix A of ‘Encapsulation Methods for Transport of Ethernet Frames Over IP/MPLS Networks’ (work-in-progress) for mapping rules of the PRI bits to PSN COS."

::= { pwVcEnetMplsPriMappingTableEntry 1 }

pwVcEnetMplsPriMappingRowStatus OBJECT-TYPE
SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Enable creating, deleting and modifying this row."
-- TBD: Need to specify exact interaction with other tables, and
-- when rows can/cannot be created/deleted/modified.
::= { pwVcEnetMplsPriMappingTableEntry 2 }

pwVcEnetMplsPriMappingStorageType OBJECT-TYPE
SYNTAX StorageType
MAX-ACCESS read-create
STATUS current
DESCRIPTION "Indicates the storage type of this row."
::= { pwVcEnetMplsPriMappingTableEntry 3 }

--
-- Ethernet PW Statistics Table
--

pwVcEnetStatsTable OBJECT-TYPE
SYNTAX SEQUENCE OF PwVcEnetStatsEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "This table contains statistical counters specific for Ethernet PW."
::= { pwVcEnetObjects 3 }
pwVcEnetStatsEntry OBJECT-TYPE
SYNTAX PwVcEnetStatsEntry
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 pwVcEnetTable."
INDEX { pwVcIndex }
::= { pwVcEnetStatsTable 1 }

PwVcEnetStatsEntry ::= SEQUENCE {
pwVcEnetStatsIllegalVlan ZeroBasedCounter32,
pwVcEnetStatsIllegalLength ZeroBasedCounter32
}

pwVcEnetStatsIllegalVlan 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 is not applicable if the PW type is ‘ethernet’
(i.e. raw mode), and should be return the value of zero in
this case."
::= { pwVcEnetStatsEntry 1 }

pwVcEnetStatsIllegalLength 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."
::= { pwVcEnetStatsEntry 2 }

---
--- Conformance description
--- In this version of the draft, only objects level conformance is
--- defined. More detailed conformance specifications is FFS.
---

pwVcEnetGroups OBJECT IDENTIFIER ::= { pwVcEnetConformance 1 }
pwVcEnetCompliances OBJECT IDENTIFIER ::= { pwVcEnetConformance 2 }

Zelig et al Expires June 2003 [page 15]
Ethernet Pseudo Wire (PW)  February 2004
Management Information Base

pwVcEnetModuleCompliance MODULE-COMPLIANCE
  STATUS  current
  DESCRIPTION
    "The compliance statement for agent that support Ethernet PW."

MODULE  -- this module
  MANDATORY-GROUPS { pwVcEnetGroup,
    pwVcStatsGroup
  }

GROUP pwVcEnetMplsPriGroup
  DESCRIPTION
    "Collection of objects defining classification to different PW based on the user’s PRI bits mapping. This group is optional, and should be implemented only for MPLS PSN where only L-LSP or single OA E-LSP, exists, and different PSN COS is required based on the PRI mapping."

 ::= { pwVcEnetCompliances 1 }

-- Units of conformance.

pwVcEnetGroup OBJECT-GROUP
  OBJECTS {
    pwVcEnetVlanMode,
    pwVcEnetPortVlan,
    pwVcEnetPortIfIndex,
    pwVcEnetVclIfIndex,
    pwVcEnetRowStatus,
    pwVcEnetStorageType
  }
  STATUS  current
  DESCRIPTION
    "Collection of objects for basic Ethernet PW config."
 ::= { pwVcEnetGroups 1 }

pwVcStatsGroup OBJECT-GROUP
  OBJECTS {
    pwVcEnetStatsIllegalVlan,
    pwVcEnetStatsIllegalLength
  }
  STATUS  current
  DESCRIPTION
    "Collection of objects counting various PW level errors."
 ::= { pwVcEnetGroups 2 }

pwVcEnetMplsPriGroup OBJECT-GROUP
Objects

\begin{verbatim}
OBJECTS {
  pwVcEnetMplsPriMapping,
  pwVcEnetMplsPriMappingRowStatus,
  pwVcEnetMplsPriMappingStorageType
}
\end{verbatim}

Status current

Description

"Collection of objects defining classification to
different PW based on the user’s PRI bits mapping.
This group is optional, and should be implemented
only for MPLS PSN where only L-LSP or single OA
E-LSP exists, and different PSN COS is required
based on the PRI mapping."

 ::= { pwVcEnetGroups 3 }

8 Security considerations

It is clear that this MIB module is potentially useful for
monitoring of PW capable PEs. This MIB 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 pwVcEnetTable and pwVcEnetMplsPriMappingTable contain
  objects to configure Ethernet PW parameters on a Provider Edge
  (PE) device. 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 pwVcEnetTable and pwVcEnetMplsPriMappingTable collectively show the pseudo wire connectivity topology and its performance characteristics. 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.

9 IANA considerations

As requested in the PW TC MIB [PWTCMIB], PWE3 related standards track MIB modules should be rooted under the pwStdMIB subtree. The IANA is requested to assign { pwStdMIB 5 } to the PW ENET MIB module specified in this document.
10 References

10.1 Normative references


[PWREQ] Xiao et al, "Requirements for Pseudo Wire Emulation Edge-to-Edge (PWE3)", work-in-progress.


10.2 Informative references


11 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

12 Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other
Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.