

Network Working Group
Request for Comments: 5602
Category: Standards Track

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

Pseudowire (PW) over MPLS PSN 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 a MIB module for PW operation over Multiprotocol Label Switching (MPLS) Label Switching Routers (LSRs).

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. Terminology	3
4. Overview	3
5. Features Checklist	4
6. MIB Module Usage	5
7. PW-MPLS-STD-MIB Example	7
8. Object Definitions	8
9. Security Considerations	28
10. IANA Considerations	29
11. References	29
11.1. Normative References	29
11.2. Informative References	30

1. Introduction

This document describes a model for managing pseudowire services for transmission over different flavors of MPLS tunnels. The general PW MIB module [RFC5601] defines the parameters global to the PW regardless of the underlying Packet Switched Network (PSN) and emulated service. This document is applicable for PWs that use MPLS PSN type in the PW-STD-MIB.

This document describes the MIB objects that define pseudowire association to the MPLS PSN, in a way that is not specific to the carried service.

Together, [RFC3811] and [RFC3812] describe the modeling of an MPLS tunnel, and a tunnel's underlying cross-connects. This MIB module supports MPLS-TE PSN, non-TE MPLS PSN (an outer tunnel created by the Label Distribution Protocol (LDP) or manually), and MPLS PW label only (no outer tunnel).

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 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].

3. Terminology

This document uses terminology from the document describing the PW architecture [RFC3985], [RFC3916], and [RFC4447].

The terms "outbound" and "inbound" in this MIB module are based on the common practice in the MPLS standards; i.e. "outbound" is toward the PSN. However, where these terms are used in an object name, the object description clarifies the exact packet direction to prevent confusion with these terms in other documents.

"PSN tunnel" is a general term indicating a virtual connection between the two Pseudowire Emulation Edge-to-Edge (PWE3) edge devices. Each tunnel may potentially carry multiple PWs inside. An MPLS tunnel is within the scope of this document.

This document uses terminology from the document describing the MPLS architecture [RFC3031] for MPLS PSN. A Label Switched Path (LSP) is modeled as described in [RFC3811] and [RFC3812] via a series of cross-connects through one or more Label Switching Routers (LSRs).

In MPLS PSN, a PW connection typically uses a PW label within a tunnel label [RFC4447]. Multiple pseudowires each with a unique PW label can share the same tunnel. For PW transport over MPLS, the tunnel label is known as the "outer" label, while the PW label is known as the "inner" label. An exception to this is with adjacent LSRs or the use of a Penultimate Hop Popping (PHP). In this case, there is an option for PWs to connect directly without an outer label.

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].

4. Overview

The MIB module structure for defining a PW service consists 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 structure 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 PW over MPLS PSN.

[RFC5542] defines some of the object types used in these modules.

5. Features Checklist

The PW-MPLS-STD-MIB module is designed to satisfy the following requirements and constraints:

- The MIB module supports both manually configured and signaled PWs.
- The MIB module supports point-to-point PW connections.
- The MIB module enables the use of any emulated service.
- The MIB module supports MPLS-TE outer tunnel, non-TE MPLS outer tunnel (an outer tunnel signaled by LDP or set up manually), and no outer tunnel (where the PW label is the only label in the MPLS stack). The latter case is applicable for manual configuration of PW over a single hop, as for signaled MPLS PSN even across a single hop there is an MPLS tunnel -- even though the actual packet may not contain the MPLS tunnel label due to PHP.

The MIB module uses Textual Conventions (TCs) from [RFC2578], [RFC2579], [RFC2580], [RFC2863], [RFC3811], [RFC3813], [RFC5542], and [RFC5601].

6. MIB Module Usage

- The PW table (pwTable) in [RFC5601] is used for all PW types (ATM, FR, Ethernet, SONET, etc.). This table contains high-level generic parameters related to the PW creation. The operator or the agent creates a row for each PW.
- If the selected PSN type in the pwTable is MPLS, the agent creates a row in the MPLS-specific parameters table (pwMplsTable) in this module, which contains MPLS-specific parameters such as EXP bits handling and outer tunnel configuration.
- The operator configures the association to the desired MPLS tunnel (required for MPLS-TE tunnels or for manually configured PWs) through the pwMplsTeOutboundTable. For the LDP-based outer tunnel, there is no need for manual configuration since there is only a single tunnel toward the peer.
- The agent creates rows in the MPLS mapping table in order to allow quick retrieval of information based on the tunnel indexes.

The relation to the MPLS network is by configuration of the edge LSR only -- i.e., the LSR that provides the PW function. Since tunnels are unidirectional, a pair of tunnels MUST exist (one for inbound, one for outbound). Figure 1 depicts a PW that originates and terminates at LSR-M. It uses tunnels A and B formed by cross-connects (XCs) Ax and Bx continuing through LSR-N to LSR-P. The concatenations of XCs create the tunnels. Note: 'X' denotes a tunnel's cross-connect.

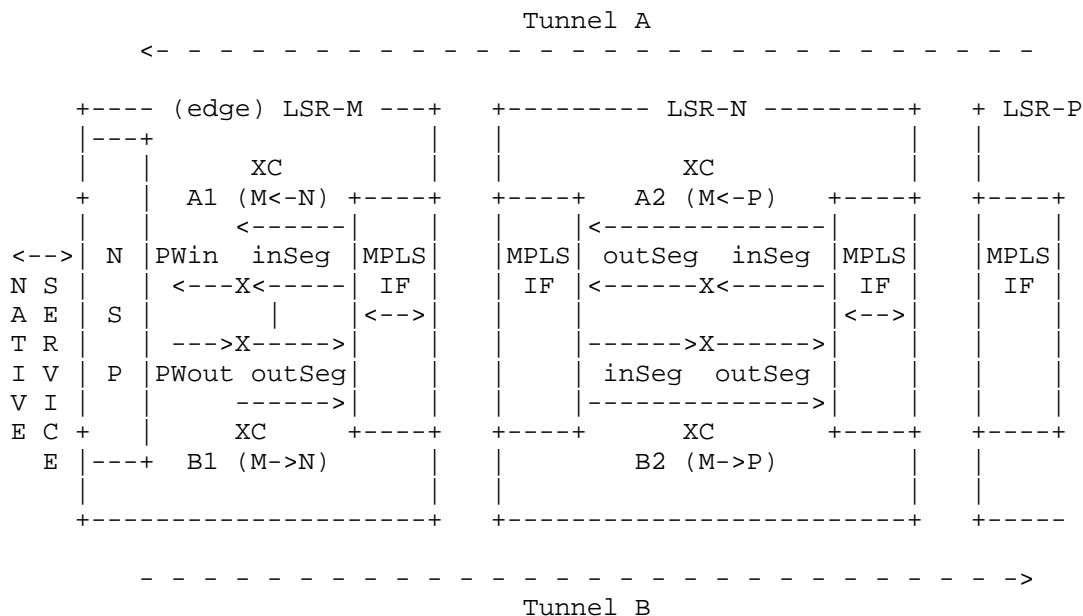


Figure 1: PW modeling over MPLS

The PW-MPLS-STD-MIB supports three options for an MPLS network:

- (1) In the MPLS-TE case, tunnels A and B are created via the MPLS-TE-STD-MIB [RFC3812]. The tunnels are associated (in each peer independently) to the PW by the four indexes that uniquely identify the tunnel at the MPLS-TE-STD-MIB.
- (2) In the non-TE case, tunnels A1 and B1 are either manually configured or set up with LDP. The tunnels are associated to the PW by the XC index in the MPLS-LSR-STD-MIB [RFC3813].
- (3) In the PW-label-only case, there is no outer tunnel on top of the PW label. This case is useful in the case of adjacent Provider Edges (PEs) in manual configuration mode. Note that for signaled tunnels, when LSR-N acts as PHP for the outer tunnel label, there are still entries for the outer tunnel in the relevant MPLS MIB modules, so even for the case of adjacent LSRs, the relevant mode is either MPLS-TE or non-TE.

A combination of MPLS-TE outer tunnel(s) and LDP outer tunnel for the same PW is allowed through the `pwMplsOutboundTunnel`. The current tunnel that is used to forward traffic is indicated in the object `pwMplsOutboundTunnelTypeInUse`.

The PW-MPLS-STD-MIB module reports through the inbound table the XC entry in the LDP-STD-MIB [RFC3815] of the PW that was signaled through LDP.

This MIB module assumes that a PW can be associated to one MPLS-TE tunnel at a time. This tunnel may be composed of multiple instances (i.e., LSP), each represented by a separate instance index. The selection of the active LSP out of the possible LSPs in the tunnel is out of the scope of this MIB module as it is part of the MPLS PSN functionality. The current active LSP is reported through this MIB module.

It is important to note that inbound (tunnel originated in the remote PE) mapping is not configured or reported through the PW-MPLS-STD-MIB module since the local PE does not know the inbound association between specific PW and MPLS tunnels.

7. PW-MPLS-STD-MIB Example

The following example (supplement the example provided in [RFC5601]) assumes that the node has already established the LDP tunnel to the peer node and that a PW has been configured in the pwTable in [RFC5601] with pwPsnType equal 'mpls'.

The agent creates an entry in pwMplsTable with the following parameters:

```

pwMplsMplsType          mplsNonTe(1), -- LDP tunnel
pwMplsExpBitsMode       outerTunnel(1), -- Default
pwMplsExpBits           0, -- Default
pwMplsTtl               2, -- Default
pwMplsLocalLdpID        192.0.2.200:0,
pwMplsLocalLdpEntityIndex 1,
pwMplsPeerLdpID         192.0.2.5:0,
pwMplsStorageType       nonVolatile(3)

```

The agent also creates an entry in pwMplsOutboundTable for reporting the mapping of the PW on the LDP tunnel:

```

pwMplsOutboundLsrXcIndex 100, - The XC number for the
                          -- LDP tunnel
pwMplsOutboundTunnelIndex 0, -- No TE tunnel
pwMplsOutboundTunnelInstance 0, -- No TE tunnel
pwMplsOutboundTunnelLclLSR 0, -- No TE tunnel
pwMplsOutboundTunnelPeerLSR 0, -- No TE tunnel
pwMplsOutboundIfIndex     0, -- Not applicable
pwMplsOutboundTunnelTypeInUse mplsNonTe(3)

```

The agent now creates entries for the PW in the following tables:

- pwMplsInboundTable
- pwMplsNonTeMappingTable (2 entries)

To create an MPLS-TE tunnel to carry this PW, the operator takes the following steps:

- Set pwMplsMplsType in pwMplsTable to both mplsNonTe(1) and mplsTe(0).
- Set pwMplsOutboundTunnelIndex, pwMplsOutboundTunnelInstance, pwMplsOutboundTunnelLclLSR, and pwMplsOutboundTunnelPeerLSR in pwMplsOutboundTable to the MPLS-TE tunnel that will carry this PW.

The agent will report the tunnel that the PW is currently using through pwMplsOutboundTunnelTypeInUse, and will report the PW to MPLS-TE tunnel/LSP mapping in pwMplsTeMappingTable.

8. Object Definitions

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

IMPORTS

MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, mib-2
FROM SNMPv2-SMI -- [RFC2578]

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

StorageType
FROM SNMPv2-TC -- [RFC2579]

InterfaceIndexOrZero
FROM IF-MIB -- [RFC2863]

MplsTunnelIndex, MplsTunnelInstanceIndex,
MplsLdpIdentifier, MplsLsrIdentifier
FROM MPLS-TC-STD-MIB -- [RFC3811]

MplsIndexType
FROM MPLS-LSR-STD-MIB -- [RFC3813]

PwIndexType
FROM PW-TC-STD-MIB -- [RFC5542]


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

pwMplsStdMIB MODULE-IDENTITY
LAST-UPDATED "200906120000Z" -- 12 June 2009 00:00:00 GMT
ORGANIZATION "Pseudowire Emulation Edge-to-Edge (PWE3) Working
Group."
CONTACT-INFO
"
    David Zelig, Editor
    Email: davidz@corrigent.com

    Thomas D. Nadeau, Editor
    Email: tom.nadeau@bt.com

    The PWE3 Working Group (email distribution pwe3@ietf.org,
    http://www.ietf.org/html.charters/pwe3-charter.html)
"
DESCRIPTION
"This MIB module complements the PW-STD-MIB module for PW
operation over MPLS.

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 5602;
see the RFC itself for full legal notices.

"

-- Revision history.

REVISION "200906120000Z" -- 12 June 2009 00:00:00 GMT

DESCRIPTION

"First published as RFC 5602. "

::= { mib-2 181 }

-- Top-level components of this MIB.

-- Notifications

pwMplsNotifications OBJECT IDENTIFIER

::= { pwMplsStdMIB 0 }

-- Tables, Scalars

pwMplsObjects OBJECT IDENTIFIER

::= { pwMplsStdMIB 1 }

-- Conformance

pwMplsConformance OBJECT IDENTIFIER

::= { pwMplsStdMIB 2 }

-- PW MPLS table

pwMplsTable OBJECT-TYPE

SYNTAX SEQUENCE OF PwMplsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table controls MPLS-specific parameters when the PW is
going to be carried over MPLS PSN."

::= { pwMplsObjects 1 }

pwMplsEntry OBJECT-TYPE

SYNTAX PwMplsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A row in this table represents parameters specific to MPLS PSN for a pseudowire (PW). The row is created automatically by the local agent if the pwPsnType is mpls(1). It is indexed by pwIndex, which uniquely identifies a singular PW. 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 were found to be nonconsistent after reboot, the PW pwOperStatus MUST be declared as down(2). Any read-write object in this table MAY be changed at any time; however, change of some objects (for example, pwMplsMplsType) during PW forwarding state MAY cause traffic disruption."

INDEX { pwIndex }

::= { pwMplsTable 1 }

```
PwMplsEntry ::= SEQUENCE {
    pwMplsMplsType          BITS,
    pwMplsExpBitsMode       INTEGER,
    pwMplsExpBits           Unsigned32,
    pwMplsTtl               Unsigned32,
    pwMplsLocalLdpID        MplsLdpIdentifier,
    pwMplsLocalLdpEntityIndex Unsigned32,
    pwMplsPeerLdpID         MplsLdpIdentifier,
    pwMplsStorageType       StorageType
}
```

pwMplsMplsType OBJECT-TYPE

```
SYNTAX BITS {
    mplsTe      (0),
    mplsNonTe   (1),
    pwOnly      (2)
}
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is set by the operator to indicate the outer tunnel types, if existing. mplsTe(0) is used if the outer tunnel is set up by MPLS-TE, and mplsNonTe(1) is used if the outer tunnel is set up by LDP or manually. A combination of mplsTe(0) and mplsNonTe(1) MAY exist. pwOnly(2) is used if there is no outer tunnel label, i.e.,

in static provisioning without an MPLS tunnel. pwOnly(2) cannot be combined with mplsNonTe(1) or mplsTe(0). An implementation that can identify automatically that the peer node is directly connected MAY support the bit pwOnly(2) as read-only.

```
"
DEFVAL { { mplsNonTe } }
::= { pwMplsEntry 1 }
```

pwMplsExpBitsMode OBJECT-TYPE

```
SYNTAX      INTEGER {
    outerTunnel      (1),
    specifiedValue   (2),
    serviceDependant (3)
}
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is set by the operator to determine the PW shim label EXP bits. The value of outerTunnel(1) is used where there is an outer tunnel -- pwMplsMplsType equals to mplsTe(0) or mplsNonTe(1). Note that in this case, there is no need to mark the PW label with the EXP bits, since the PW label is not visible to the intermediate nodes. If there is no outer tunnel, specifiedValue(2) SHOULD be used to indicate that the value is specified by pwMplsExpBits. Setting serviceDependant(3) indicates that the EXP bits are set based on a rule that is implementation specific."

```
DEFVAL { outerTunnel }
::= { pwMplsEntry 2 }
```

pwMplsExpBits OBJECT-TYPE

```
SYNTAX      Unsigned32 (0..7)
```

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is set by the operator if pwMplsExpBitsMode is set to specifiedValue(2) to indicate the MPLS EXP bits to be used on the PW shim label. Otherwise, it SHOULD be set to zero."

```
DEFVAL { 0 }
::= { pwMplsEntry 3 }
```

pwMplsTtl OBJECT-TYPE

```
SYNTAX      Unsigned32 (0..255)
```

MAX-ACCESS read-write

STATUS current
DESCRIPTION
 "This object is set by the operator to indicate the PW TTL
 value to be used on the PW shim label."
DEFVAL { 2 }
::= { pwMplsEntry 4 }

pwMplsLocalLdpID OBJECT-TYPE
SYNTAX MplsLdpIdentifier
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "The LDP identifier of the LDP entity that creates
 this PW in the local node. As the PW labels are always
 set from the per-platform label space, the last two octets
 in the LDP ID MUST always both be zeros."
REFERENCE
 "'LDP specifications', RFC 3036, section 2.2.2."
::= { pwMplsEntry 5 }

pwMplsLocalLdpEntityIndex OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "The local node LDP Entity Index of the LDP entity creating
 this PW."
::= { pwMplsEntry 6 }

pwMplsPeerLdpID OBJECT-TYPE
SYNTAX MplsLdpIdentifier
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "The peer LDP identifier of the LDP session. This object
 SHOULD return the value zero if LDP is not used or if the
 value is not yet known."
::= { pwMplsEntry 7 }

pwMplsStorageType OBJECT-TYPE
SYNTAX StorageType
MAX-ACCESS read-write
STATUS current
DESCRIPTION
 "This variable indicates the storage type for this row."
DEFVAL { nonVolatile }
::= { pwMplsEntry 8 }

-- End of PW MPLS Table

-- Pseudowire MPLS Outbound Tunnel Table

pwMplsOutboundTable OBJECT-TYPE
SYNTAX SEQUENCE OF PwMplsOutboundEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"This table reports and configures the current outbound MPLS
tunnels (i.e., toward the PSN) or the physical interface in
the case of a PW label only that carries the PW traffic. It
also reports the current outer tunnel and LSP that forward
the PW traffic."
::= { pwMplsObjects 2 }

pwMplsOutboundEntry OBJECT-TYPE
SYNTAX PwMplsOutboundEntry
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
"A row in this table configures the outer tunnel used for
carrying the PW traffic toward the PSN.
In the case of PW label only, it configures the interface
that will carry the PW traffic.

An entry in this table augments the pwMplsEntry, and is
created automatically when the corresponding row has been
created by the agent in the pwMplsEntry.

This table points to the appropriate MPLS MIB module:

In the MPLS-TE case, the three objects relevant to the
indexing of a TE tunnel head-end (as used in the
MPLS-TE-STD-MIB) are to be configured, and the tunnel
instance indicates the LSP that is currently in use for
forwarding the traffic.

In the case of signaled non-TE MPLS (an outer tunnel label
assigned by LDP), the table points to the XC entry in the
LSR-STD-MIB. If the non-TE MPLS tunnel is manually
configured, the operator configures the XC pointer to this
tunnel.

In the case of PW label only (no outer tunnel), the ifIndex
of the port to carry the PW is configured here.

It is possible to associate a PW to one TE tunnel head-end and a non-TE tunnel together. An indication in this table will report the currently active one. In addition, in the TE case, the table reports the active tunnel instance (i.e., the specific LSP in use). Any read-write object in this table MAY be changed at any time; however, change of some objects (for example, MPLS-TE indexes) during PW forwarding state MAY cause traffic disruption."

AUGMENTS { pwMplsEntry }

::= { pwMplsOutboundTable 1 }

```
PwMplsOutboundEntry ::= SEQUENCE {
    pwMplsOutboundLsrXcIndex      MplsIndexType,
    pwMplsOutboundTunnelIndex     MplsTunnelIndex,
    pwMplsOutboundTunnelInstance  MplsTunnelInstanceIndex,
    pwMplsOutboundTunnelLclLSR    MplsLsrIdentifier,
    pwMplsOutboundTunnelPeerLSR   MplsLsrIdentifier,
    pwMplsOutboundIfIndex         InterfaceIndexOrZero,
    pwMplsOutboundTunnelTypeInUse INTEGER
}
```

pwMplsOutboundLsrXcIndex OBJECT-TYPE

SYNTAX MplsIndexType

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is applicable if the pwMplsMplsType mplsNonTe(1) bit is set, and MUST return a value of zero otherwise. If the outer tunnel is signaled, the object is read-only and indicates the XC index in the MPLS-LSR-STD-MIB of the outer tunnel toward the peer. Otherwise (tunnel is set up manually), the operator defines the XC index of the manually created outer tunnel through this object.

"

::= { pwMplsOutboundEntry 1 }

pwMplsOutboundTunnelIndex OBJECT-TYPE

SYNTAX MplsTunnelIndex

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is applicable if the pwMplsMplsType mplsTe(0) bit is set, and MUST return a value of zero otherwise. It is part of the set of indexes for the outbound tunnel.

The operator sets this object to represent the desired tunnel head-end toward the peer for carrying the PW traffic.

"

::= { pwMplsOutboundEntry 2 }

pwMplsOutboundTunnelInstance OBJECT-TYPE

SYNTAX MplsTunnelInstanceIndex

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object is applicable if the pwMplsMplsType mplsTe(0) bit is set, and MUST return a value of zero otherwise. It indicates the actual tunnel instance that is currently active and carrying the PW traffic. It SHOULD return the value zero if the information from the MPLS-TE application is not yet known.

"

::= { pwMplsOutboundEntry 3 }

pwMplsOutboundTunnelLclLSR OBJECT-TYPE

SYNTAX MplsLsrIdentifier

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is applicable if the pwMplsMplsType mplsTe(0) bit is set, and MUST return a value of all zeros otherwise. It is part of the set of indexes for the outbound tunnel. The operator sets this object to represent the desired tunnel head-end toward the peer for carrying the PW traffic.

"

::= { pwMplsOutboundEntry 4 }

pwMplsOutboundTunnelPeerLSR OBJECT-TYPE

SYNTAX MplsLsrIdentifier

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object is applicable if the pwMplsMplsType mplsTe(0) bit is set, and MUST return a value of zero otherwise. It is part of the set of indexes for the outbound tunnel. Note that in most cases, it equals to pwPeerAddr.

"

::= { pwMplsOutboundEntry 5 }

pwMplsOutboundIfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero


```

MAX-ACCESS      read-write
STATUS          current
DESCRIPTION
    "This object is applicable if the pwMplsMplsType pwOnly(0)
    bit is set, and MUST return a value of zero otherwise.
    The operator configures the ifIndex of the outbound port
    in this case.
    "
 ::= { pwMplsOutboundEntry 6 }

```

pwMplsOutboundTunnelTypeInUse OBJECT-TYPE

```

SYNTAX          INTEGER {
                    notYetKnown (1),
                    mplsTe       (2),
                    mplsNonTe    (3),
                    pwOnly       (4)
                }
MAX-ACCESS      read-only
STATUS          current
DESCRIPTION
    "This object indicates the current tunnel that is carrying
    the PW traffic.
    The value of notYetKnown(1) should be used if the agent is
    currently unable to determine which tunnel or interface is
    carrying the PW, for example, because both tunnels are in
    operational status down.
    "
 ::= { pwMplsOutboundEntry 7 }

```

-- End of PW MPLS Outbound Tunnel table

-- PW MPLS inbound table

pwMplsInboundTable OBJECT-TYPE

```

SYNTAX          SEQUENCE OF PwMplsInboundEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION
    "This table indicates the PW LDP XC entry in the
    MPLS-LSR-STD-MIB for signaled PWs.
    "
 ::= { pwMplsObjects 3 }

```

pwMplsInboundEntry OBJECT-TYPE

```

SYNTAX          PwMplsInboundEntry
MAX-ACCESS      not-accessible
STATUS          current
DESCRIPTION

```

"A row in this table is created by the agent for each signaled PW, and shows the XC index related to the PW signaling in the inbound direction in the MPLS-LSR-STD-MIB that controls and display the information for all the LDP signaling processes in the local node.

"
INDEX { pwIndex }

::= { pwMplsInboundTable 1 }

PwMplsInboundEntry ::= SEQUENCE {
 pwMplsInboundXcIndex MplsIndexType
}

pwMplsInboundXcIndex OBJECT-TYPE

SYNTAX MplsIndexType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The XC index representing this PW in the inbound direction. It MUST return the value zero if the information is not yet known."

::= { pwMplsInboundEntry 1 }

-- End of PW MPLS inbound table

-- PW to Non-TE mapping Table.

pwMplsNonTeMappingTable OBJECT-TYPE

SYNTAX SEQUENCE OF PwMplsNonTeMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table indicates the PW association to the outbound tunnel in non-TE applications, maps the PW to its (inbound) XC entry, and indicates the PW-to-physical interface mapping for a PW without an outer tunnel.

"

::= { pwMplsObjects 4 }

pwMplsNonTeMappingEntry OBJECT-TYPE

SYNTAX PwMplsNonTeMappingEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A row in this table displays the association between the PW and
- its non-TE MPLS outbound outer tunnel,

- its XC entry in the MPLS-LSR-STD-MIB, or
- its physical interface if there is no outer tunnel (PW label only) and manual configuration.

Rows are created in this table by the agent depending on the setting of pwMplsMplsType:

- If the pwMplsMplsType mplsNonTe(1) bit is set, the agent creates a row for the outbound direction (pwMplsNonTeMappingDirection set to psnBound(1)). The pwMplsNonTeMappingXcIndex holds the XC index in the MPLS-LSR-STD-MIB of the PSN-bound outer tunnel. pwMplsNonTeMappingIfIndex MUST be zero for this row.

- If the pwMplsMplsType pwOnly(2) bit is set, the agent creates a row for the outbound direction (pwMplsNonTeMappingDirection set to psnBound(1)). The pwMplsNonTeMappingIfIndex holds the ifIndex of the physical port this PW will use in the outbound direction. pwMplsNonTeMappingXcIndex MUST be zero for this row.

- If the PW has been set up by a signaling protocol (i.e., pwOwner equal pwIdFecSignaling(2) or genFecSignaling(3)), the agent creates a row for the inbound direction (pwMplsNonTeMappingDirection set to fromPsn(2)). The pwMplsNonTeMappingXcIndex holds the XC index in the MPLS-LSR-STD-MIB of the PW LDP-generated XC entry. pwMplsNonTeMappingIfIndex MUST be zero for this row.

An application can use this table to quickly retrieve the PW carried over specific non-TE MPLS outer tunnel or physical interface.

"

```
INDEX { pwMplsNonTeMappingDirection,
        pwMplsNonTeMappingXcIndex,
        pwMplsNonTeMappingIfIndex,
        pwMplsNonTeMappingPwIndex }
```

```
::= { pwMplsNonTeMappingTable 1 }
```

```
PwMplsNonTeMappingEntry ::= SEQUENCE {
    pwMplsNonTeMappingDirection    INTEGER,
    pwMplsNonTeMappingXcIndex      MplsIndexType,
    pwMplsNonTeMappingIfIndex      InterfaceIndexOrZero,
    pwMplsNonTeMappingPwIndex      PwIndexType
}
```

pwMplsNonTeMappingDirection OBJECT-TYPE

```

SYNTAX      INTEGER {
                psnBound (1),
                fromPsn  (2)
            }
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION

```

```

    "Index for the conceptual XC row identifying the tunnel-to-PW
    mappings, indicating the direction of the packet flow for
    this entry.
    psnBound(1) indicates that the entry is related to
    packets toward the PSN.
    fromPsn(2) indicates that the entry is related to
    packets coming from the PSN.
    "

```

```

 ::= { pwMplsNonTeMappingEntry 1 }

```

pwMplsNonTeMappingXcIndex OBJECT-TYPE

```

SYNTAX      MplsIndexType
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION

```

```

    "See the description clause of pwMplsNonTeMappingEntry for
    the usage guidelines of this object."

```

```

 ::= { pwMplsNonTeMappingEntry 2 }

```

pwMplsNonTeMappingIfIndex OBJECT-TYPE

```

SYNTAX      InterfaceIndexOrZero
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION

```

```

    "See the description clause of pwMplsNonTeMappingEntry for
    the usage guidelines of this object."

```

```

 ::= { pwMplsNonTeMappingEntry 3 }

```

pwMplsNonTeMappingPwIndex OBJECT-TYPE

```

SYNTAX      PwIndexType
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION

```

```

    "The value that represents the PW in the pwTable."

```

```

 ::= { pwMplsNonTeMappingEntry 4 }

```

```

-- End of PW to Non-TE mapping Table.

```

```

-- PW to TE MPLS tunnels mapping Table.

```

```

pwMplsTeMappingTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF PwMplsTeMappingEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "This table reports the PW association to the
         outbound MPLS tunnel for MPLS-TE applications."
    ::= { pwMplsObjects 5 }

pwMplsTeMappingEntry OBJECT-TYPE
    SYNTAX          PwMplsTeMappingEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A row in this table represents the association
         between a PW and its MPLS-TE outer (head-end) tunnel.

        An application can use this table to quickly retrieve the
        list of the PWs that are configured on a specific MPLS-TE
        outer tunnel.

        The pwMplsTeMappingTunnelInstance reports the actual
        LSP out of the tunnel head-end that is currently
        forwarding the traffic.

        The table is indexed by the head-end indexes of a TE
        tunnel and the PW index.
        "

    INDEX { pwMplsTeMappingTunnelIndex,
            pwMplsTeMappingTunnelInstance,
            pwMplsTeMappingTunnelPeerLsrID,
            pwMplsTeMappingTunnelLocalLsrID,
            pwMplsTeMappingPwIndex }

    ::= { pwMplsTeMappingTable 1 }

PwMplsTeMappingEntry ::= SEQUENCE {
    pwMplsTeMappingTunnelIndex      MplsTunnelIndex,
    pwMplsTeMappingTunnelInstance   MplsTunnelInstanceIndex,
    pwMplsTeMappingTunnelPeerLsrID  MplsLsrIdentifier,
    pwMplsTeMappingTunnelLocalLsrID MplsLsrIdentifier,
    pwMplsTeMappingPwIndex          PwIndexType
}

```

pwMplsTeMappingTunnelIndex OBJECT-TYPE

SYNTAX MplsTunnelIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Primary index for the conceptual row identifying the MPLS-TE tunnel that is carrying the PW traffic."

::= { pwMplsTeMappingEntry 1 }

pwMplsTeMappingTunnelInstance OBJECT-TYPE

SYNTAX MplsTunnelInstanceIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object identifies the MPLS-TE LSP that is carrying the PW traffic. It MUST return the value zero if the information of the specific LSP is not yet known. Note that based on the recommendation in the MPLS-TC-STD-MIB, instance index 0 should refer to the configured tunnel interface."

::= { pwMplsTeMappingEntry 2 }

pwMplsTeMappingTunnelPeerLsrID OBJECT-TYPE

SYNTAX MplsLsrIdentifier

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object identifies the peer LSR when the outer tunnel is MPLS-TE."

::= { pwMplsTeMappingEntry 3 }

pwMplsTeMappingTunnelLocalLsrID OBJECT-TYPE

SYNTAX MplsLsrIdentifier

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object identifies the local LSR."

::= { pwMplsTeMappingEntry 4 }

pwMplsTeMappingPwIndex OBJECT-TYPE

SYNTAX PwIndexType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object returns the value that represents the PW in the pwTable."

::= { pwMplsTeMappingEntry 5 }

```
-- End of PW to TE MPLS tunnels mapping Table.

-- conformance information

pwMplsGroups      OBJECT IDENTIFIER ::= { pwMplsConformance 1 }
pwMplsCompliances OBJECT IDENTIFIER ::= { pwMplsConformance 2 }

-- Compliance requirement for fully compliant implementations.

pwMplsModuleFullCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The compliance statement for agents that provide full
        support for the PW-MPLS-STD-MIB module.  Such devices
        can then be monitored and also be configured using
        this MIB module."

    MODULE      -- this module
    MANDATORY-GROUPS { pwMplsGroup,
                        pwMplsOutboundMainGroup,
                        pwMplsInboundGroup,
                        pwMplsMappingGroup
                      }

    GROUP      pwMplsOutboundTeGroup
    DESCRIPTION "This group MUST be supported if the implementation
                allows MPLS-TE tunnels to carry PW traffic.
                "

    OBJECT      pwMplsMplsType
    DESCRIPTION "Support of pwOnly(2) is not required.  At least one
                of mplsTe(0) or mplsNonTe(1) MUST be supported if
                signaling of PW is supported.
                "

    OBJECT      pwMplsExpBitsMode
    DESCRIPTION "Support of specifiedValue(2) and
                serviceDependant(3) is optional.
                "

    OBJECT      pwMplsLocalLdpID
    MIN-ACCESS   read-only
    DESCRIPTION "A read-write access is required if the
                implementation supports more than one LDP entity
                identifier for PW signaling.
                "

    OBJECT      pwMplsLocalLdpEntityIndex
```

MIN-ACCESS read-only
 DESCRIPTION "A read-write access is required if the
 implementation supports more than one LDP entity
 index for PW signaling.
 "

OBJECT pwMplsOutboundLsrXcIndex
 MIN-ACCESS read-only
 DESCRIPTION "A value other than zero MUST be supported if the
 implementation supports non-TE signaling of the
 outer tunnel.
 A read-write access MUST be supported if the
 implementation supports PW label manual setting
 and carrying them over non-TE tunnels.
 "

OBJECT pwMplsOutboundIfIndex
 MIN-ACCESS read-only
 DESCRIPTION "A value other than zero and read-write operations
 MUST be supported if the implementation supports
 manually configured PW without MPLS outer tunnel.
 "

::= { pwMplsCompliances 1 }

-- Compliance requirement for Read Only compliant implementations.

pwMplsModuleReadOnlyCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for agents that provide read-
 only support for the PW-MPLS-STD-MIB module. Such
 devices can then be monitored but cannot be configured
 using this MIB module."

MODULE -- this module

MANDATORY-GROUPS { pwMplsGroup,
 pwMplsOutboundMainGroup,
 pwMplsInboundGroup,
 pwMplsMappingGroup
 }

GROUP pwMplsOutboundTeGroup

DESCRIPTION "This group MUST be supported if the implementation
 allows MPLS-TE tunnels to carry PW traffic.
 "

OBJECT pwMplsMplsType

MIN-ACCESS read-only

DESCRIPTION "Write access is not required.
Support of pwOnly(2) is not required. At least one
of mplTe(0) or mplNonTe(1) MUST be supported if
signaling of PW is supported.
"

OBJECT pwMplsExpBitsMode
MIN-ACCESS read-only
DESCRIPTION "Write access is not required.
Support of specifiedValue(2) and serviceDependant(3)
is optional.
"

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

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

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

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

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

OBJECT pwMplsOutboundLsrXcIndex
MIN-ACCESS read-only
DESCRIPTION "Write access is not required.
A value other than zero MUST be supported if the
implementation supports non-TE signaling of the
outer tunnel.
"

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

```

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

```

```

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

```

```

OBJECT      pwMplsOutboundIfIndex
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required.
            A value other than zero MUST be supported if the
            implementation supports manually configured PW
            without MPLS outer tunnel.
"

```

```
 ::= { pwMplsCompliances 2 }
```

```
-- Units of conformance.
```

```
pwMplsGroup OBJECT-GROUP
```

```

OBJECTS {
    pwMplsMplsType,
    pwMplsExpBitsMode,
    pwMplsExpBits,
    pwMplsTtl,
    pwMplsLocalLdpID,
    pwMplsLocalLdpEntityIndex,
    pwMplsPeerLdpID,
    pwMplsStorageType
}

```

```
STATUS current
```

```
DESCRIPTION
```

```
    "Collection of objects needed for PW over MPLS PSN
    configuration."
```

```
 ::= { pwMplsGroups 1 }
```

```
pwMplsOutboundMainGroup OBJECT-GROUP
```

```

OBJECTS {
    pwMplsOutboundLsrXcIndex,
    pwMplsOutboundIfIndex,
    pwMplsOutboundTunnelTypeInUse
}

```

```
STATUS current
```

```
DESCRIPTION
```

```
    "Collection of objects needed for outbound association of
      PW and MPLS tunnel."
 ::= { pwMplsGroups 2 }
```

pwMplsOutboundTeGroup OBJECT-GROUP

```
  OBJECTS {
    pwMplsOutboundTunnelIndex,
    pwMplsOutboundTunnelInstance,
    pwMplsOutboundTunnelLclLSR,
    pwMplsOutboundTunnelPeerLSR
  }
```

STATUS current

DESCRIPTION

```
    "Collection of objects needed for outbound association of
      PW and MPLS-TE tunnel."
 ::= { pwMplsGroups 3 }
```

pwMplsInboundGroup OBJECT-GROUP

```
  OBJECTS {
    pwMplsInboundXcIndex
  }
```

STATUS current

DESCRIPTION

```
    "Collection of objects needed for inbound PW presentation.
      This group MUST be supported if PW signaling through LDP is
      used."
 ::= { pwMplsGroups 4 }
```

pwMplsMappingGroup OBJECT-GROUP

```
  OBJECTS {
    pwMplsNonTeMappingPwIndex,
    pwMplsTeMappingPwIndex
  }
```

STATUS current

DESCRIPTION

```
    "Collection of objects needed for mapping association of
      PW and MPLS tunnel."
 ::= { pwMplsGroups 5 }
```

END

9. Security Considerations

It is clear that this MIB module is potentially useful for monitoring 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 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 pwMplsTable, pwMplsNonTeMappingTable and pwMplsTeMappingTable collectively contain objects to provision PW over MPLS tunnels. 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:

- o the pwMplsTable, pwMplsNonTeMappingTable, pwMplsTeMappingTable and pwMplsOutboundTable collectively show the PW over MPLS association. 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.

10. IANA Considerations

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

Descriptor	OBJECT IDENTIFIER value
-----	-----
pwMplsStdMIB	{ mib-2 181 }

11. References

11.1. Normative References

- [BCP14] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "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. Kastenholtz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

- [RFC3811] Nadeau, T., Ed., and J. Cucchiara, Ed., "Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management", RFC 3811, June 2004.
- [RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004.
- [RFC3813] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)", RFC 3813, June 2004.
- [RFC4447] Martini, L., Ed., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.
- [RFC5542] Nadeau, T., Ed., Zelig, D., Ed., and O. Nicklass, Ed., "Definitions of Textual Conventions for Pseudowire (PW) Management", RFC 5542, May 2009.
- [RFC5601] Nadeau, T., Ed. and D. Zelig, Ed. "Pseudowire (PW) Management Information Base (MIB)", RFC 5601, July 2009.

11.2. 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.
- [RFC3815] Cucchiara, J., Sjostrand, H., and J. Luciani, "Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)", RFC 3815, June 2004.
- [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.

Authors' Addresses

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

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

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

EMail: tom.nadeau@bt.com

