Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 5, 2009

E. Beili
Actelis Networks
M. Morgenstern
ECI Telecom
N. Nair
Wipro Technologies

September 01, 2008

xDSL multi-pair bonding (G.Bond) MIB
draft-ietf-adslmib-gbond-mib-02.txt

Status of This Memo

Abstract

This document defines Management Information Base (MIB) module for use with network management protocols in TCP/IP-based internets. This document proposes an extension to the Interfaces Group MIB with a set of common objects for managing multi-pair bonded Digital Subscriber Line (xDSL) interfaces, defined in ITU-T recommendations G.998.1, G.998.2 and G.998.3. The MIB modules specific to each bonding technology are defined in GBOND-ATM-MIB, GBOND-ETH-MIB and GBOND-TDIM-MIB respectively.

Table of Contents

1. Introduction

The xDSL Multi-Pair Bonding, allows a service provider to provide high bandwidth services to business and residential customers over multiple xDSL lines, with greater speed and resiliency, than the service over a single xDSL line, bridging the gap between xDSL and fiber-based transport.

There are three xDSL Multi-Pair Bonding schemes, also known under collective name G.Bond:

- The ATM-Based Multi-Pair Bonding, specified in ITU-T G.998.1 recommendation [G.998.1], which defines a method for bonding (or aggregating) of multiple xDSL lines (or individual bearer channels in multiple xDSL lines) into a single bi-directional logical link carrying an ATM stream. This specification can be viewed as an evolution of the legacy Inverse Multiplexing over ATM (IMA) technology [af-phy-0086], applied to xDSL with variable rates on each line/bearer channel.

- The Ethernet-Based Multi-Pair Bonding, specified in ITU-T G.998.2 recommendation [G.998.2], which defines a method for bonding (or aggregating) of multiple xDSL lines (or individual bearer channels in multiple xDSL lines) into a single bi-directional logical link carrying an Ethernet stream. This specification can be viewed as IEEE 802.3-2005 [802.3] Clause 61 Physical Medium Entity (PME) Aggregation, generalized to work over any xDSL technology. (2Base-TL and 10Pass-TS interfaces defined by IEEE use G.SHDSL and VDSL technology respectively).

- The Multi-pair bonding using time-division inverse multiplexing (TDIM), specified in ITU-T G.998.3 recommendation [G.998.3], which defines a method for bonding (or aggregating) of multiple xDSL lines into a single bi-directional logical link carrying a mix of various traffic streams (e.g., Ethernet, ATM, TDM).

Architecturally all three bonding schemes define a new "bonded" Transport Protocol Specific - Transmission Convergence (TPS-TC) sub-layer, stacked above multiple ATM-TC, Ethernet/PTM-TC or STM-TC (clear channel) sub-layers for the ATM, Ethernet or TDIM bonding respectively. Each underlying TPS-TC sub-layer represents a protocol specific gamma-interface to an xDSL line or an individual bearer channel of an xDSL line. Bonding of multiple bearer channels in the same xDSL line is not allowed.

All schemes allow bonding of up to 32 individual line/channel sub-layers with variable rates, providing common functionality for the configuration, initialization, operation and monitoring of the bonded

link.

This document defines a MIB module common to all 3 schemes. Additional managed objects, specific to each bonding technology, are defined in GBOND-ATM-MIB [I-D.ietf-adslmib-gbond-atm-mib], GBOND-ETH-MIB [I-D.ietf-adslmib-gbond-eth-mib] and GBOND-TDIM-MIB [I-D.ietf-adslmib-gbond-tdim-mib] modules.

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 document 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].

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 [RFC2119].

3. The DSL Forum Management Framework for xDSL Bonding

This document makes use of the DSL Forum technical report Management Framework for xDSL Bonding [WT-159], defining a management model and a hierarchy of management objects for the bonded xDSL interfaces.

4. Relationship to Other MIB modules

This section outlines the relationship of the MIB modules defined in this document with other MIB modules described in the relevant RFCs. Specifically, the following MIB modules are discussed: Interfaces Group MIB (IF-MIB), Inverse Stack Table MIB (IF-INVERTED-STACK-MIB) Interface Stack Capability MIB (IF-CAP-STACK-MIB), G.Bond scheme specific modules: G.Bond/ATM (GBOND-ATM-MIB), G.Bond/Ethernet (GBOND-ETH-MIB) and G.Bond/TDIM (GBOND-TDIM-MIB), and DSL specific MIB modules: ADSL (ADSL-LINE-EXT-MIB), ADSL2 (ADSL2-LINE-MIB), SHDSL (HDSL2-SHDSL-LINE-MIB), VDSL (VDSL-LINE-MIB) and VDSL2 (VDSL2-LINE-MIB).

4.1. Relationship to Interfaces Group MIB module

A bonded xDSL port is a stacked (a.k.a. aggregated or bonded) interface and as such is managed using generic interface management objects defined in the IF-MIB [RFC2863].

The stack management, i.e., actual connection of the sub-layers to the top layer interface, is done via the ifStackTable, as defined in the IF-MIB [RFC2863] and its inverse ifInvStackTable, as defined in the IF-INVERTED-STACK-MIB [RFC2864].

The ifCapStackTable and its inverse ifInvCapStackTable defined in the IF-CAP-STACK-MIB [RFC5066], extend the stack management with an ability to describe possible connections or cross-connect capability, when a flexible cross-connect matrix is present between the interface layers.

4.1.1. Layering Model

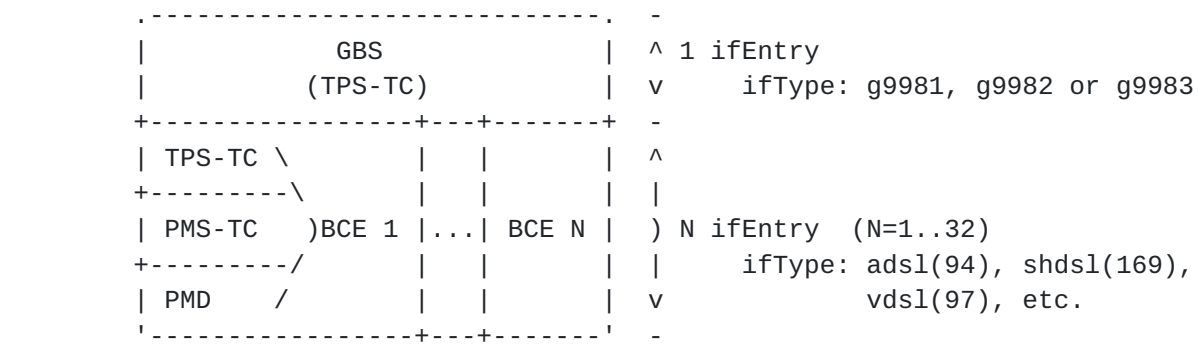
A G.Bond interface can aggregate up to 32 channel sub-layers, with each channel representing an xDSL line or an xDSL bearer channel. For the purpose of brevity we will refer to the bonded interface as Generic Bonded Sub-layer (GBS) and to the channel sub-layer as Bonding Channel Entity (BCE).

A generic G.Bond device can have a number of GBS ports, each connected to a particular upper layer (e.g., Media Access Control (MAC) interface for G.998.2 scheme), while simultaneously cross-connected to a number of underlying BCEs, with a single GBS per BCE relationship.

A GBS port is represented in the Interface table (ifTable) as a separate interface with an ifType of g9981, g9982 or g9983 for a particular bonding scheme.

Each BCE in the aggregated GBS port is represented in the ifTable as a separate interface with an ifType relevant to a particular xDSL technology, e.g., shdsl(169) or vdsl(97). The ifType values are defined in [IANAifType-MIB].

The following figure shows the layering diagram and corresponding use of ifTable for the bonded xDSL interfaces:



BCE - Bonding Channel Entity
GBS - Generic Bonded Sub-layer
PMD - Physical Medium Dependent
TPS-TC - Transport Protocol Specific - Transmission Convergence
PMS-TC - Physical Media Specific - Transmission Convergence

Figure 1: Use of ifTable for bonded xDSL interfaces

The ifStackTable is indexed by the ifIndex values of the aggregated G.Bond port (GBS) and the BCEs connected to it. ifStackTable allows a Network Management application to determine which BCEs are connected to a particular GBS and change connections (if supported by the application). The ifInvStackTable, being an inverted version of the ifStackTable, provides an efficient means for a Network Management application to read a subset of the ifStackTable and thereby determine which GBS runs on top of a particular BCE.

The ifCapStackTable defined in the IF-CAP-STACK-MIB module, specifies for each higher-layer interface (e.g., GBS port) a list of lower-layer interfaces (e.g., BCEs), which can possibly be cross-connected to that higher-layer interface, determined by the cross-connect capability of the device. This table, modeled after ifStackTable, is read only, reflecting current cross-connect capability of a stacked interface, which can be dynamic in some implementations (e.g., if xDSL lines are located on a pluggable module and the module is pulled out). Note that BCE availability per GBS, described by ifCapStackTable, can be constrained by other parameters, for example by aggregation capacity of a GBS or by the BCE in question being already connected to another GBS. So, in order to ensure that a particular BCE can be connected to the GBS, all respective parameters (e.g., ifCapStackTable, ifStackTable and gBondCapacity) SHALL be inspected.

The ifInvCapStackTable, also defined in the IF-CAP-STACK-MIB module,

describes which higher-layer interfaces (e.g., GBS ports) can possibly be connected to a particular lower-layer interface (e.g., BCE), providing inverted mapping of ifCapStackTable. While it contains no additional information beyond that already contained in the ifCapStackTable, the ifInvCapStackTable has the ifIndex values in its INDEX clause in the reverse order, i.e., the lower-layer interface first, and the higher-layer interface second, providing an efficient means for a Network Management application to read a subset of the ifCapStackTable and thereby determine which interfaces can be connected to run on top of a particular interface.

4.1.2. G.Bond Aggregation Function (GAF)

The G.Bond Aggregation Function (GAF) allows a number of BCEs to be aggregated onto a single logical GBS port, by splitting the incoming traffic into multiple streams, transmitting each stream over a specific BCE and combining the streams at the remote GBS port, preserving the original traffic order.

Big Ethernet frames MAY be fragmented before the transmission and reassembled at the remote end to minimize transportation delay.

GAF is OPTIONAL, meaning that a device with a single BCE MAY perform fragmentation and re-assembly if this function is supported by the device. The agent is REQUIRED to report on the GAF capability for all types of G.Bond ports (ATM, Ethernet and TDIM).

The GBOND-MIB module allows a Network Management application to query GAF capability and enable/disable it if supported. Note that enabling GAF effectively turns on fragmentation and re-assembly, even on a single-BCE port.

4.1.3. Discovery Operation

The G.Bond ports may optionally support discovery operation, whereby BCEs, during initialization, exchange information about their respective aggregation groups (GBS), via xDSL handshake protocol. This information can then be used to detect copper misconnections or for an automatic assignment of the local BCEs into aggregation groups instead of a fixed pre-configuration.

The MIB module defined in this document allow a Network Management application to control G.Bond Discovery mechanism and query its results. Note that the Discovery mechanism can work only if GAF is supported and enabled.

Two tables are used by the G.Bond Discovery mechanism: ifStackTable and ifCapStackTable. The following pseudo-code gives an example of

the Discovery and automatic BCE assignment for a generic multi-GBS G.Bond device, located at Central Office (CO), using objects defined in this MIB module, IF-CAP-STACK-MIB and IF-MIB modules [Note that automatic BCE assignment is only shown here for the purposes of the example. Fixed BCE pre-assignment, manual assignment or auto-assignment using an alternative internal algorithm may be chosen by a particular implementation]:

```
// Go over all GBS ports in the CO device
FOREACH gbs[i] IN CO_device
{ // Perform discovery and auto-assignment on GBS ports
  // with room for more Channels
  IF ( gbs[i].NumBCEs &lt; gbs[i].BondCapacity )
  { // Assign a unique 6-octets local discovery code to the GBS
    // e.g., MAC address
    dc = gbs[i].DiscoveryCode = MAC[i];
    // Go over all disconnected Channels, which can
    // pottentially be connected to the GBS
    FOREACH bce[j] IN ifCapStackTable[gbs[i]] AND
                  NOT IN ifStackTable[gbs[i]]  // not connected
    { // Try to grab the remote RT_device, by writing the value
      // of the local 6 Byte discovery code to the remote
      // discovery code register (via handshake mechanism).
      // This operation is atomic Set-if-Clear action, i.e., it
      // would succeed only if the remote discovery register was
      // zero. Read the remote discovery code register via Get
      // operation to see if the RT_device, attached via the BCE
      // is indeed marked as being the CO_device peer.
      bce[j].RemoteDiscoveryCode = dc;          // Set-if-Clear
      r = bce[j].RemoteDiscoveryCode;           // Get
      IF ( r == dc AND gbs[i].NumBCEs &lt; gbs[i].BondCapacity)
      { // Remote RT_device connected via BCE[j] is/was a peer
        // for GBS[i] and there room for another BCE in the
        // GBS[i] aggregation group (max. Bonding capacity is
        // not reached yet).
        // Connect this BCE to the GBS (via ifStackTable,
        // ifInvStackTable being inverse of ifStackTable is
        // updated automatically, i.e., gbs[i] is auto-added
        // to ifInvStackTable[bce[j]])
        ADD bce[j] TO ifStackTable[gbs[i]];
        gbs[i].NumBCEs = gbs[i].NumBCEs + 1;
        // Discover all other disconnected BCEs,
        // attached to the same RT_device and connect them to
        // the GBS provided there is enough room for more BCEs.
        FOREACH bce[k] IN ifCapStackTable[gbs[i]] and
                      NOT IN ifStackTable[gbs[i]]
        { // Get Remote Discovery Code from the BCE to see if
          // it belongs to a connected RT_device "grabbed" by
```

```
            // the CO_device.
            r = bce[k].RemoteDiscoveryCode;
            IF ( r == dc AND gbs[i].NumBCEs &lt; gbs[i].BondCapacity)
            { // Physically connect the BCE to the GBS
              // (gbs[i] is auto-added TO ifInvStackTable[bce[k]])
              ADD bce[k] TO ifStackTable[gbs[i]];
              gbs[i].NumBCEs = gbs[i].NumBCEs + 1;
            }
          }
        }
        // At this point we have discovered all local BCEs which
        // are physically connected to the same remote RT_device
        // and connected them to GBS[i]. Go to the next GBS.
        BREAK;
      }
    }
  }
```

An SNMP Agent for a G.Bond device builds ifCapStackTable and its inverse ifInvCapStackTable on device initialiation, according to the cross-connect capabilities of the device.

Adding a BCE to the ifStackTable row for a specific GBS, involves actual connection of the BCE to the GBS.

Note that GBS port does not have to be operationally 'down' for the connection to succeed. In fact, a dynamic BCE addition (and removal) MAY be implemented with an available BCE being initialized first (by setting its ifAdminStatus to 'up') and then added to an operationally 'up' GBS port, by modifying a respective ifStackTable (and respective ifInvStackTable) entry.

It is RECOMMENDED that a removal of the last operationally 'up' BCE from an operationally 'up' GBS would be rejected by the implementation, as this action would completely drop the link.

In addition to the standard handshake-based discovery described above, [G.998.2] defines an optional frame-based discovery and pair management. These frame-based methods are discussed in [I-D.ietf-adslmib-gbond-eth-mib].

4.1.4. G.Bond ports initialization

G.Bond ports being built on top of xDSL technology, require a lengthy initialization or 'training' process, before any data can pass. During this initialization both ends of a link (peers) work cooperatively to achieve required data rate on a particular copper pair. Sometimes, when the copper line is too long or the noise on

the line is too high, that 'training' process may fail to achieve a specific target rate with required characteristics.

The ifAdminStatus object from the IF-MIB, controls the desired state of a GBS with all the BCEs connected to it or of an individual BCE port. Setting this object to 'up' instructs a particular GBS or a BCE to start initialization process, which may take tens of seconds for G.Bond ports. The ifOperStatus object shows the operational state of an interface (extended by ifMauMediaAvailable object from MAU-MIB [RFC4836] for GBS and *Status object from a relevant line MIB for BCE interfaces).

A disconnected BCE may be initialized by changing the ifAdminState from 'down' to 'up'. Changing the ifAdminState to 'up' on the GBS initializes all BCEs connected to that particular GBS. Note that in case of bonding some interfaces may fail to initialize while others succeed. The GBS is considered operationally 'up' if at least one bonded BCE is operationally 'up'. When all BCEs connected to the GBS are 'down' the GBS SHALL be considered operationally 'lowerLayerDown'. The GBS SHALL be considered operationally 'notPresent' if it is not connected to any BCE. The GBS/BCE interface SHALL remain operationally 'down' during initialization.

4.1.5. Usage of ifTable

Both BCE and GBS interfaces are managed using interface specific management objects defined in the GBOND-MIB module and generic interface objects from the ifTable of IF-MIB, with all management table entries referenced by the interface index ifIndex.

The following table summarizes G.Bond specific interpretations for some of the ifTable objects specified by the mandatory ifGeneralInformationGroup:

IF-MIB object	G.Bond interpretation
ifIndex	Interface index. Note that each BCE and each GBS in the G.Bond PHY MUST have a unique index, as there some GBS and BCE specific attributes accessible only on the GBS or BCE level.
ifType	g9981, g9982 or g9982 for the ATM, Ethernet or TDIM GBS respectively, shdsl(169) for G.SHDSL BCE, vdsl(97) for VDSL BCE etc.

ifSpeed	Operating data rate for the BCE. For the GBS it is the sum of the current operating data rates of all BCEs in the aggregation group, without the encapsulation overhead and G.Bond overhead, but accounting for the Inter-Frame Gaps (IFG). When a GBS or a BCE is operating in an assymetrical fashion (upstream data rate differs from the downstream one) the lowest of the values is shown.
ifAdminStatus	Setting this object to 'up' instructs a particular GBS (with all BCEs connected to it) or a BCE to start initialization process
ifOperStatus	a relevant *Status object from a particular line MIB supplements the 'down' value of ifOperStatus for BCEs.

Table 1: G.Bond interpretation of IF-MIB objects

4.2. Relationship to xDSL MIB modules

Each xDSL technology is described in a relevant xDSL line MIB module: e.g., HDSL2-SHDSL-LINE-MIB [RFC4319] for G.SHDSL, ADSL-LINE-EXT-MIB [RFC3440] for ADSL, ADSL2-LINE-MIB [RFC4706] for ADSL2, VDSL-LINE-MIB [RFC3728] for VDSL or VDSL2-LINE-MIB [I-D.ietf-adslmib-vdsl2-mib] for VDSL2.

These MIBs are used to manage individual xDSL lines/channels (BCEs).

5. MIB Structure

5.1. Overview

The main management objects defined in the GBOND-MIB module are split into 2 groups:

- gBondPort - containing objects for configuration, capabilities, status and notifications, common to all G.Bond ports (GBS).

- gBondBce - containing a single common object for configuration of the remote discovery code per BCE. Note that the rest of the objects for BCE configuration, capabilities, status and notifications, is located in relevant xDSL line MIB modules as well as in the bonding scheme specific MIB modules.

5.2. Mapping of DSL Forum WT-159 Managed Objects

This section contains the mapping between relevant managed objects (attributes) defined in [WT-159] and managed objects defined in this document and in associated MIB modules, i.e., the IF-MIB [RFC2863].

G.Bond Managed Object	Corresponding SNMP Object
oBondingGroup - Basic Package (Mandatory)	
aGroupID	ifIndex (IF-MIB)
aGroupBondSchemesSupported	gBondSchemesSupported
aGroupBondPeerSchemesSupported	gBondPeerSchemesSupported
aGroupAdminBondScheme	gBondAdminScheme
aGroupOperBondScheme	gBondOperScheme
aGroupPeerOperBondScheme	gBondPeerOperScheme
aGroupEnd	gBondPortSide
aGroupOperState	ifOperStatus (IF-MIB)
aGroupAdminState	ifAdminStatus (IF-MIB)
aGroupStatus	gBondFltStatus
aGroupCapacity	gBondCapacity
aGroupPeerCapacity	gBondPeerCapacity
aGroupNumChannels	gBondNumBCEs
aGroupName	ifName (IF-MIB)
aGroupDiscoveryCode	gBondDiscoveryCode
aGroupUpRate	gBondUpDataRate
aGroupDownRate	gBondDownDataRate
aGroupTargetUpRate	gBondTargetUpDataRate
aGroupTargetDownRate	gBondTargetDownDataRate
aGroupThreshLowUpRate	gBondThreshLowUpRate
aGroupThreshLowDownRate	gBondThreshLowDownRate

aGroupLowRateCrossingEnable	gBondLowRateCrossingEnable
nGroupLowUpRateCrossing	gBondLowUpRateCrossing
nGroupLowDownRateCrossing	gBondLowDownRateCrossing
aGroupLinkUpDownEnable	_EdNote: Currently IF-MIB doesn't provide a control for the linkUp/linkDown notifications. Can we define a control in one MIB module while the notifications are in another?_
nGroupLinkUp	linkDown (IF-MIB)
nGroupLinkDown	linkUp (IF-MIB)
oLine - Basic Package (Mandatory)	
aLineID	ifIndex (IF-MIB)
aLineType	ifType (IF-MIB)
aLineOperState	ifOperStatus (IF-MIB)
aLineStatus	*dsl*CurrStatus (*DSL-LINE-MIB)
aLineEnd	*dsl*Side (*DSL-LINE-MIB)
aLineAdminState	ifAdminStatus (IF-MIB)
aLineRemoteDiscoveryCode	gBondBceRemoteDiscoveryCode
aLineUpDownEnable	_EdNote: Currently IF-MIB doesn't provide a control for the linkUp/linkDown notifications. Can we define a control in one MIB module while the notifications are in another?_
nLineUp	linkUp (IF-MIB)
nLineDown	linkDown (IF-MIB)

Table 2: Mapping of WT-159 Managed Objects

6. xDSL Multi-pair Bonding MIB Definitions

```
GBOND-MIB DEFINITIONS ::= BEGIN

  IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
    Unsigned32, Gauge32, mib-2
      FROM SNMPv2-SMI         -- [RFC2578]
    TEXTUAL-CONVENTION, TruthValue, PhysAddress
      FROM SNMPv2-TC          -- [RFC2579]
    MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
      FROM SNMPv2-CONF        -- [RFC2580]
    ifIndex
      FROM IF-MIB             -- [RFC2863]
    ;

  gBondMIB MODULE-IDENTITY
    LAST-UPDATED "200809010000Z"  -- Sep 01, 2008
    ORGANIZATION "IETF ADSL MIB Working Group"
    CONTACT-INFO
      "WG charter:
        http://www.ietf.org/html.charters/adslmib-charter.html

      Mailing Lists:
        General Discussion: adslmib@ietf.org
        To Subscribe: adslmib-request@ietf.org
        In Body: subscribe your_email_address

       Chair: Menachem Dodge
      Postal: ECI Telecom, Ltd.
              30 Hasivim St.,
              Petach-Tikva 49517
              Israel
       Phone: +972-3-926-8421
       EMail: menachem.dodge@ecitele.com

      Editor: Edward Beili
      Postal: Actelis Networks, Inc.
              25 Bazel St., P.O.B. 10173
              Petach-Tikva 10173
              Israel
       Phone: +972-3-924-3491
       EMail: edward.beili@actelis.com

      Editor: Moti Morgenstern
      Postal: ECI Telecom
```

30 Hasivim St.
Petach-Tikva 49517
Israel
Phone: +972-3-926-6258
EMail: moti.morgenstern@ecitele.com

Editor: Narendranath Nair
Posatl: Wipro Technologies
Keonics Electronics City
Bangalore 560 100
India
Phone: +91-80-2852-0408 x85338
EMail: narendranath.nair@wipro.com"

DESCRIPTION
"The objects in this MIB module are used to manage the multi-pair bonded xDSL Interfaces, defined in ITU-T recommendations G.998.1, G.998.2 and G.998.3.

This MIB module MUST be used in conjunction with a bonding scheme specific MIB module, that is, GBOND-ATM-MIB, GBOND-ETH-MIB or GBOND-TDIM-MIB.

The following references are used throughout this MIB module:

[G.998.1] refers to:
ITU-T Recommendation G.998.1: 'ATM-based multi-pair bonding', January 2005.

[G.998.2] refers to:
ITU-T Recommendation G.998.2: 'Ethernet-based multi-pair bonding', January 2005.

[G.998.3] refers to:
ITU-T Recommendation G.998.3: 'Multi-pair bonding using time-division inverse multiplexing', January 2005.

[WT-159] refers to:
DSL Forum Technical Report: 'Management Framework for xDSL Bonding', September 2008.

Naming Conventions:
BCE - Bonding Channel Entity
CO - Central Office
CPE - Customer Premises Equipment
GBS - Generic Bonding Sublayer
SNR - Signal to Noise Ratio

```
        Copyright (C) The IETF Trust (2008).
        This version of this MIB module is part of RFC XXXX;
        see the RFC itself for full legal notices."

      REVISION    "200809010000Z"  -- Sep 01, 2008
      DESCRIPTION "Initial version, published as RFC XXXX."

        -- EdNote: Replace XXXX with the actual RFC number &
        -- remove this note

      ::= { mib-2 ZZZ }

        -- EdNote: Replace ZZZ with a real OID once it is
        -- allocated & remove this note.

    -- Sections of the module
    -- Structured as recommended by [RFC4181], Appendix D

    gBondObjects     OBJECT IDENTIFIER ::= { gBondMIB 1 }

    gBondConformance OBJECT IDENTIFIER ::= { gBondMIB 2 }

    -- Groups in the module

    gBondPort        OBJECT IDENTIFIER ::= { gBondObjects 1 }

    gBondBce         OBJECT IDENTIFIER ::= { gBondObjects 2 }

    -- Textual Conventions

    GBondSchemeList ::= TEXTUAL-CONVENTION
      STATUS       current
      DESCRIPTION
        "This textual convention defines a bitmap of possible ITU-T
        G.998 (G.Bond) bonding schemes. Currently there are 3 bonding
        schemes defined: G.998.1, G.998.2 and G.998.3, identified by
        bit values g9981(1), g9982(2) and g9983(3), respectively.
        An additional value of unknown(0), can be returned as a result
        of GET operation, when an value of the object cannot be
        determined, for example a peer GBS cannot be reached or it
        does not support any kind of bonding."
      SYNTAX       BITS {
        unknown(0),
        g9981(1),
        g9982(2),
        g9983(3)
      }
```

```
      GBondScheme ::= TEXTUAL-CONVENTION
        STATUS       current
        DESCRIPTION
          "This textual convention defines ITU-T G.998 bonding scheme
          values. Possible values are:
            unknown(0) - undefined or unknown
            g9981(1)   - G.998.1 (G.Bond/ATM)
            g9982(2)   - G.998.2 (G.Bond/Ethernt)
            g9983(3)   - G.998.3 (G.Bond/TDIM)."
        SYNTAX       INTEGER {
          unknown(0),
          g9981(1),
          g9982(2),
          g9983(3)
        }


     -- Port Notifications Group

      gBondPortNotifications OBJECT IDENTIFIER ::= { gBondPort 0 }

      gBondLowUpRateCrossing NOTIFICATION-TYPE
        OBJECTS {
          -- ifIndex is not needed here since we are under specific GBS
          gBondUpDataRate,
          gBondThreshLowUpRate
        }
        STATUS      current
        DESCRIPTION
          "This notification indicates that the G.Bond port' upstream
          data rate has reached/dropped below or exceeded the low
          upstream rate threshold, specified by gBondThreshLowUpRate.

          This notification MAY be sent for the -O subtype ports
          while the port is up, on the crossing event in both
          directions: from normal (rate is above the threshold) to low
          (rate equals the threshold or below it) and from low to
          normal. This notification is not applicable to the -R
          subtypes.

          It is RECOMMENDED that a small debouncing period of 2.5 sec,
          between the detection of the condition and notification,
          is implemented to prevent simultaneous LinkUp/LinkDown and
          gBondLowUpRateCrossing notifications to be sent.

          The adaptive nature of the G.Bond technology allows the port
          to adapt itself to the changes in the copper environment,
          e.g., an impulse noise, alien crosstalk, or a
```

```
          micro-interruption may temporarily drop one or more BCEs in
          the aggregation group, causing a rate degradation of the
          aggregated G.Bond link. The dropped BCEs would then try to
          re-initialize, possibly at a lower rate than before, adjusting
          the rate to provide required target SNR margin.

          Generation of this notification is controlled by the
          gBondLowRateCrossingEnable object.

          This object maps to the WT-159 notification
          nGroupLowUpRateCrossing."
        REFERENCE
          "[WT-159] 5.5.1.24"
        ::= { gBondPortNotifications 1 }

      gBondLowDownRateCrossing NOTIFICATION-TYPE
        OBJECTS {
          -- ifIndex is not needed here since we are under specific GBS
          gBondDownDataRate,
          gBondThreshLowDownRate
        }
        STATUS      current
        DESCRIPTION
          "This notification indicates that the G.Bond port' downstream
          data rate has reached/dropped below or exceeded the low
          downstream rate threshold, specified by
          gBondThreshLowDownRate.

          This notification MAY be sent for the -O subtype ports
          while the port is up, on the crossing event in both
          directions: from normal (rate is above the threshold) to low
          (rate equals the threshold or below it) and from low to
          normal. This notification is not applicable to the -R
          subtypes.

          It is RECOMMENDED that a small debouncing period of 2.5 sec,
          between the detection of the condition and notification,
          is implemented to prevent simultaneous LinkUp/LinkDown and
          gBondLowDownRateCrossing notifications to be sent.

          The adaptive nature of the G.Bond technology allows the port
          to adapt itself to the changes in the copper environment,
          e.g., an impulse noise, alien crosstalk, or a
          micro-interruption may temporarily drop one or more BCEs in
          the aggregation group, causing a rate degradation of the
          aggregated G.Bond link. The dropped BCEs would then try to
          re-initialize, possibly at a lower rate than before,
          adjusting the rate to provide required target SNR margin.
```

```
          Generation of this notification is controlled by the
          gBondLowRateCrossingEnable object.

          This object maps to the WT-159 notification
          nGroupLowDownRateCrossing."
        REFERENCE
          "[WT-159] 5.5.1.25"
        ::= { gBondPortNotifications 2}

      -- G.Bond Port (GBS) group

      gBondPortConfTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF GBondPortConfEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "Table for Configuration of G.Bond GBS ports. Entries in this
          table MUST be maintained in a persistent manner"
        ::= { gBondPort 1 }

      gBondPortConfEntry OBJECT-TYPE
        SYNTAX      GBondPortConfEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "An entry in the G.Bond Port Configuration table.
          Each entry represents an G.Bond port indexed by the ifIndex.
          Note that an G.Bond GBS port runs on top of a single
          or multiple BCE port(s), which are also indexed by ifIndex."
        INDEX  { ifIndex }
        ::= { gBondPortConfTable 1 }

      GBondPortConfEntry ::=
        SEQUENCE {
          gBondAdminScheme              GBondScheme,
          gBondPeerAdminScheme          GBondScheme,
          gBondDiscoveryCode            PhysAddress,
          gBondTargetUpDataRate         Unsigned32,
          gBondTargetDownDataRate       Unsigned32,
          gBondThreshLowUpRate          Unsigned32,
          gBondThreshLowDownRate        Unsigned32,
          gBondLowRateCrossingEnable    TruthValue
        }

      gBondAdminScheme  OBJECT-TYPE
        SYNTAX      GBondScheme
        MAX-ACCESS  read-write
        STATUS      current
```

```
        DESCRIPTION
          "A desired bonding scheme for a G.Bond GBS port.
          The following values instruct the port to use corresponding
          bonding scheme if supported:
            g9981(1)      - instructs the port to use G.998.1 bonding
            g9982(2)      - instructs the port to use G.998.2 bonding
            g9983(3)      - instructs the port to use G.998.3 bonding

          Changing of the gBondAdminScheme MUST be performed when the
          link is Down. Attempts to change this object MUST be rejected
          (in case of SNMP with the error inconsistentValue), if the
          link is Up or Initializing. Attempts to change this object to
          an unsupported bonding scheme (see gBondSchemesSupported)
          SHALL be rejected (in case of SNMP with the error wrongValue).

          This object MUST be maintained in a persistent manner.

          This object maps to the WT-159 attribute
          aGroupAdminBondScheme."
        REFERENCE
          "[WT-159] 5.5.1.6"
        ::= { gBondPortConfEntry 1 }

      gBondPeerAdminScheme  OBJECT-TYPE
        SYNTAX      GBondScheme
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
          "A desired bonding scheme for a peer (link partner) G.Bond
          port (GBS).
          The following values instruct the peer port to use
          corresponding bonding scheme if supported:
            g9981(1)      - instructs the port to use G.998.1 bonding
            g9982(2)      - instructs the port to use G.998.2 bonding
            g9983(3)      - instructs the port to use G.998.3 bonding

          Changing of the gBondAdminScheme MUST be performed when the
          link is Down. Attempts to change this object MUST be rejected
          (in case of SNMP with the error inconsistentValue), if the
          link is Up or Initializing. Attempts to change this object to
          an unsupported bonding scheme (see gBondPeerSchemesSupported)
          SHALL be rejected (in case of SNMP with the error wrongValue).

          This object MUST be maintained in a persistent manner.

          This object maps to the WT-159 attribute
          aGroupPeerAdminBondScheme."
        REFERENCE
```

```
        "[WT-159] 5.5.1.7"
      ::= { gBondPortConfEntry 2 }

    gBondDiscoveryCode  OBJECT-TYPE
      SYNTAX      PhysAddress (SIZE(6))
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
        "A Discovery Code of the G.Bond port (GBS).
        A unique 6 octet long code used by the Discovery function.
        This object MUST be instantiated for the -O subtype GBS before
        writing operations on the gBondBceRemoteDiscoveryCode
        (Set_if_Clear and Clear_if_Same) are performed by BCEs
        associated with the GBS.
        The initial value of this object for -R subtype ports after
        reset is all zeroes. For -R subtype ports, the value of this
        object cannot be changed directly. This value may be changed
        as a result of writing operation on the
        gBondBceRemoteDiscoveryCode object of remote BCE of -O
        subtype, connected to one of the local BCEs associated with
        the GBS.

        Discovery MUST be performed when the link is Down.
        Attempts to change this object MUST be rejected (in case of
        SNMP with the error inconsistentValue), if the link is Up or
        Initializing.

        This object maps to the WT-159 attribute
        aGroupDiscoveryCode."
      REFERENCE
        "[WT-159] 5.5.1.20; [802.3] 61.2.2.8.3, 61.2.2.8.4,
        45.2.6.6.1, 45.2.6.8, 61A.2"
      ::= { gBondPortConfEntry 3 }

    gBondTargetUpDataRate  OBJECT-TYPE
      SYNTAX      Unsigned32(1..1000000|9999999)
      UNITS       "Kbps"
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
        "A desired G.Bond port Data Rate in the upstream direction,
        in Kbps, to be achieved during initialization, under
        restrictions placed upon the member BCEs by their respective
        configuration settings.
        This object represents a sum of individual BCE upstream data
        rates, modified to compensate for fragmentation and
        encapsulation overhead (e.g., for an Ethernet service, the
        target data rate of 10Mbps SHALL allow lossless transmission
```

```
          of full-duplex 10Mbps Ethernet frame stream with minimal
          inter-frame gap).
          Note that the target upstream data rate may not be achieved
          during initialization (e.g., due to unavailability of required
          BCEs) or the initial bandwidth could deteriorate, so that the
          actual upstream data rate (gBondUpDataRate) could be less
          than gBondTargetUpDataRate.

          The value between 1 and 1000000 indicates that the total
          upstream data rate of the G.Bond port after initialization
          SHALL be equal to the target data rate or less, if the target
          upstream data rate cannot be achieved under the restrictions
          configured for BCEs. In case the copper environment allows to
          achieve higher upstream data rate than that specified by this
          object, the excess capability SHALL be either converted to
          additional SNR margin or reclaimed by minimizing transmit
          power.

          The value of 9999999 means that the target data rate is not
          fixed and SHALL be set to the maximum attainable rate during
          initialization (Best Effort), under specified spectral
          restrictions and with desired SNR Margin per BCE.

          This object is read-write for the -O subtype G.Bond ports
          and irrelevant for the -R subtypes.

          Changing of the Target Upstream Data Rate MUST be performed
          when the link is Down. Attempts to change this object MUST be
          rejected (In case of SNMP with the error inconsistentValue),
          if the link is Up or Initializing.

          This object MUST be maintained in a persistent manner.

          This object maps to the WT-159 attribute
          aGroupTargetUpRate."
        REFERENCE
          "[WT-159] 5.5.1.17"
        ::= { gBondPortConfEntry 4 }

      gBondTargetDownDataRate  OBJECT-TYPE
        SYNTAX      Unsigned32(1..1000000|9999999)
        UNITS       "Kbps"
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
          "A desired G.Bond port Data Rate in the downstream direction,
          in Kbps, to be achieved during initialization, under
          restrictions placed upon the member BCEs by their respective
```

```
          configuration settings.
          This object represents a sum of individual BCE downstream data
          rates, modified to compensate for fragmentation and
          encapsulation overhead (e.g., for an Ethernet service, the
          target data rate of 10Mbps SHALL allow lossless transmission
          of full-duplex 10Mbps Ethernet frame stream with minimal
          inter-frame gap).
          Note that the target downstream data rate may not be achieved
          during initialization (e.g., due to unavailability of required
          BCEs) or the initial bandwidth could deteriorate, so that the
          actual downstream data rate (gBondDownDataRate) could be less
          than gBondTargetDownDataRate.

          The value between 1 and 1000000 indicates that the total
          downstream data rate of the G.Bond port after initialization
          SHALL be equal to the target data rate or less, if the target
          downstream data rate cannot be achieved under the restrictions
          configured for BCEs. In case the copper environment allows to
          achieve higher downstream data rate than that specified by
          this object, the excess capability SHALL be either converted
          to additional SNR margin or reclaimed by minimizing transmit
          power.

          The value of 9999999 means that the target data rate is not
          fixed and SHALL be set to the maximum attainable rate during
          initialization (Best Effort), under specified spectral
          restrictions and with desired SNR Margin per BCE.

          This object is read-write for the -O subtype G.Bond ports
          and irrelevant for the -R subtypes.

          Changing of the Target Downstream Data Rate MUST be performed
          when the link is Down. Attempts to change this object MUST be
          rejected (In case of SNMP with the error inconsistentValue),
          if the link is Up or Initializing.

          This object MUST be maintained in a persistent manner.

          This object maps to the WT-159 attribute
          aGroupTargetDownRate."
        REFERENCE
          "[WT-159] 5.5.1.18"
        ::= { gBondPortConfEntry 5 }

      gBondThreshLowUpRate  OBJECT-TYPE
        SYNTAX      Unsigned32(1..1000000)
        UNITS       "Kbps"
        MAX-ACCESS  read-write
```

```
        STATUS      current
        DESCRIPTION
          "This object configures the G.Bond port low upstream rate
          crossing alarm threshold. When the current value of
          gBondUpDataRate for this port reaches/drops below or exceeds
          this threshold, an gBondLowUpRateCrossing notification MAY be
          generated if enabled by gBondLowRateCrossingEnable.

          This object is read-write for the -O subtype G.Bond ports
          and irrelevant for the -R subtypes.

          This object MUST be maintained in a persistent manner.

          This object maps to the WT-159 attribute
          aGroupthreshLowUpRate."
        REFERENCE
          "[WT-159] 5.5.1.21"
        ::= { gBondPortConfEntry 6 }

      gBondThreshLowDownRate  OBJECT-TYPE
        SYNTAX      Unsigned32(1..1000000)
        UNITS       "Kbps"
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
          "This object configures the G.Bond port low downstream rate
          crossing alarm threshold. When the current value of
          gBondDownDataRate for this port reaches/drops below or exceeds
          this threshold, an gBondLowDownRateCrossing notification MAY
          be generated if enabled by gBondLowRateCrossingEnable.

          This object is read-write for the -O subtype G.Bond ports
          and irrelevant for the -R subtypes.

          This object MUST be maintained in a persistent manner.

          This object maps to the WT-159 attribute
          aGroupThreshDownUpRate."
        REFERENCE
          "[WT-159] 5.5.1.22"
        ::= { gBondPortConfEntry 7 }

      gBondLowRateCrossingEnable  OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
          "Indicates whether gBondLowUpRateCrossing and
```

```
          gBondLowDownRateCrossing notifications should be generated
          for this interface.

          Value of true(1) indicates that the notifications are enabled.
          Value of false(2) indicates that the notifications are
          disabled.

          This object is read-write for the -O subtype G.Bond ports
          and irrelevant for the -R subtypes.

          This object MUST be maintained in a persistent manner.

          This object maps to the WT-159 attribute
          aGroupLowRateCrossingEnable."
        REFERENCE
          "[WT-159] 5.5.1.23"
        ::= { gBondPortConfEntry 8 }


      gBondPortCapabilityTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF GBondPortCapabilityEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "Table for Capabilities of G.Bond Ports. Entries in this table
          MUST be maintained in a persistent manner"
        ::= { gBondPort 2 }

      gBondPortCapabilityEntry OBJECT-TYPE
        SYNTAX      GBondPortCapabilityEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "An entry in the G.Bond Port Capability table.
          Each entry represents an G.Bond port indexed by the ifIndex.
          Note that a G.Bond GBS port runs on top of a single
          or multiple BCE port(s), which are also indexed by ifIndex."
        INDEX  { ifIndex }
        ::= { gBondPortCapabilityTable 1 }

      GBondPortCapabilityEntry ::=
        SEQUENCE {
          gBondSchemesSupported           GBondSchemeList,
          gBondPeerSchemesSupported       GBondSchemeList,
          gBondCapacity                   Unsigned32,
          gBondPeerCapacity               Unsigned32
        }
```

```
      gBondSchemesSupported  OBJECT-TYPE
        SYNTAX      GBondSchemeList
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "Bonding Capability of the G.Bond port (GBS). This is a
          read-only bitmap of the possible bonding schemes supported by
          the GBS. The various bit-positions are:
            g9981(1)      - GBS is capable of G.998.1 bonding
            g9982(2)      - GBS is capable of G.998.2 bonding
            g9983(3)      - GBS is capable of G.998.3 bonding

          Note that for ports supporting multiple bonding schemes the
          actual administrative scheme is set via gBondAdminScheme
          object. The current operating bonding scheme is reflected in
          the gBondOperScheme.

          This object maps to the WT-159 attribute
          aGroupBondSchemesSupported."
        REFERENCE
          "[WT-159] 5.5.1.2"
        ::= { gBondPortCapabilityEntry 1 }

      gBondPeerSchemesSupported  OBJECT-TYPE
        SYNTAX      GBondSchemeList
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "Bonding Capability of the peer G.Bond port (GBS). This is a
          read-only bitmap of the possible bonding schemes supported by
          the link partner GBS. The various bit-positions are:
            unknown(0)    - GBS does not support bonding or
                            the peer unit could not be reached.
            g9981(1)      - GBS is capable of G.998.1 bonding
            g9982(2)      - GBS is capable of G.998.2 bonding
            g9983(3)      - GBS is capable of G.998.3 bonding

          Note that for ports supporting multiple bonding schemes the
          actual administrative scheme is set via gBondPeerAdminScheme
          object. The current operating bonding scheme is reflected in
          the gBondPeerOperScheme.

          This object maps to the WT-159 attribute
          aGroupBondPeerSchemesSupported."
        REFERENCE
          "[WT-159] 5.5.1.3"
        ::= { gBondPortCapabilityEntry 2 }
```

```
      gBondCapacity  OBJECT-TYPE
        SYNTAX      Unsigned32 (1..32)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "Number of BCEs that can be aggregated by the local GBS.
          The number of BCEs currently assigned to a particular G.Bond
          port (gBondNumBCEs) is never greater than gBondCapacity.

          This object maps to the WT-159 attribute aGroupCapacity."
        REFERENCE
          "[WT-159] 5.5.1.12"
        ::= { gBondPortCapabilityEntry 3 }

      gBondPeerCapacity  OBJECT-TYPE
        SYNTAX      Unsigned32 (0|1..32)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "Number of BCEs that can be aggregated by the peer GBS port.
          Value of 0 is returned when peer Bonding Capacity is unknown
          (peer cannot be reached).

          This object maps to the WT-159 attribute aGroupPeerCapacity."
        REFERENCE
          "[WT-159] 5.5.1.13"
        ::= { gBondPortCapabilityEntry 4 }


      gBondPortStatusTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF GBondPortStatusEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "This table provides overall status information of G.Bond
          ports, complementing the generic status information from the
          ifTable of IF-MIB. Additional status information about
          connected BCEs is available from the relevant line MIBs

          This table contains live data from the equipment. As such,
          it is NOT persistent."
        ::= { gBondPort 3 }

      gBondPortStatusEntry OBJECT-TYPE
        SYNTAX      GBondPortStatusEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
```

```
        "An entry in the G.Bond Port Status table.
        Each entry represents an G.Bond port indexed by the ifIndex.
        Note that an G.Bond GBS port runs on top of a single
        or multiple BCE port(s), which are also indexed by ifIndex."
      INDEX  { ifIndex }
      ::= { gBondPortStatusTable 1 }

    GBondPortStatusEntry ::=
      SEQUENCE {
        gBondOperScheme                 GBondScheme,
        gBondPeerOperScheme             GBondScheme,
        gBondUpDataRate                 Gauge32,
        gBondDownDataRate               Gauge32,
        gBondFltStatus                  BITS,
        gBondPortSide                   INTEGER,
        gBondNumBCEs                    Unsigned32
      }

    gBondOperScheme  OBJECT-TYPE
      SYNTAX      GBondScheme
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
        "Current operating bonding scheme of a G.Bond port.
        The possible values are:
          g9981(1)      - G.998.1 bonding
          g9982(2)      - G.998.2 bonding
          g9983(3)      - G.998.3 bonding

        This object maps to the WT-159 attribute
        aGroupOperBondScheme."
      REFERENCE
        "[WT-159] 5.5.1.4"
      ::= { gBondPortStatusEntry 1 }

    gBondPeerOperScheme  OBJECT-TYPE
      SYNTAX      GBondScheme
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
        "Curent operating bonding scheme of a G.Bond port link
        partner.
        The possible values are:
          unknown(0)     - peer cannot be reached due to the link
                           state
          g9981(1)      - G.998.1 bonding
          g9982(2)      - G.998.2 bonding
          g9983(3)      - G.998.3 bonding
```

```
          This object maps to the WT-159 attribute
          aGroupPeerOperBondScheme."
        REFERENCE
          "[WT-159] 5.5.1.5"
        ::= { gBondPortStatusEntry 2 }

      gBondUpDataRate  OBJECT-TYPE
        SYNTAX      Gauge32
        UNITS       "bps"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "A current G.Bond port operational Data Rate in the upstream
          direction, in bps.
          This object represents an estimation of the sum of individual
          BCE upstream data rates, modified to compensate for
          fragmentation and encapsulation overhead (e.g., for an
          Ethernet service, the target data rate of 10Mbps SHALL allow
          lossless transmission of full-duplex 10Mbps Ethernet frame
          stream with minimal inter-frame gap).

          Note that for symmetrical interfaces gBondUpDataRate ==
          gBondDownDataRate == ifSpeed.

          This object maps to the WT-159 attribute aGroupUpRate."
        REFERENCE
          "[WT-159] 5.5.1.15"
        ::= { gBondPortStatusEntry 3 }

      gBondDownDataRate  OBJECT-TYPE
        SYNTAX      Gauge32
        UNITS       "bps"
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "A current G.Bond port operational Data Rate in the downstream
          direction, in bps.
          This object represents an estimation of the sum of individual
          BCE downstream data rates, modified to compensate for
          fragmentation and encapsulation overhead (e.g., for an
          Ethernet service, the target data rate of 10Mbps SHALL allow
          lossless transmission of full-duplex 10Mbps Ethernet frame
          stream with minimal inter-frame gap).

          Note that for symmetrical interfaces gBondUpDataRate ==
          gBondDownDataRate == ifSpeed.

          This object maps to the WT-159 attribute aGroupDownRate."
```

```
        REFERENCE
          "[WT-159] 5.5.1.16"
        ::= { gBondPortStatusEntry 4 }


      gBondFltStatus  OBJECT-TYPE
        SYNTAX      BITS {
          noPeer(0),
          peerPowerLoss(1),
          peerBondSchemeMismatch(2)
          bceSubTypeMismatch(3),
          lowRate(4)
        }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "G.Bond (GBS) port Fault Status. This is a bitmap of possible
          conditions. The various bit positions are:
            noPeer              - peer GBS cannot be reached (e.g.,
                                  no BCEs attached, all BCEs are Down
                                  etc.).
            peerPowerLoss       - peer GBS has indicated impending unit
                                  failure due to loss of local power
                                  ('Dying Gasp').
            peerBondSchemeMismatch - operating bonding scheme of a peer
                                  GBS is different from the local one.
            bceSubTypeMismatch  - local BCEs in the aggregation group
                                  are not of the same sub-type, e.g.,
                                  some BCEs in the local device are -O
                                  while others are -R subtype.
            lowRate             - gBondUpRate/gBondDownRate of the port
                                  has reached or dropped below
                                  gBondThreshLowUpRate/
                                  gBondThreshLowUpRate.

          This object is intended to supplement ifOperStatus object
          in IF-MIB and ifMauMediaAvailable in MAU-MIB.

          This object maps to the WT-159 attribute aGroupStatus."
        REFERENCE
          "[WT-159] 5.5.1.9; IF-MIB, ifOperStatus; MAU-MIB,
          ifMauMediaAvailable"
        ::= { gBondPortStatusEntry 5 }

      gBondPortSide  OBJECT-TYPE
        SYNTAX      INTEGER {
          subscriber(1),
          office(2),
```

```
          unknown(3)
        }
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "G.Bond port mode of operation (subtype).
          The value of 'subscriber' indicates the port is designated as
          '-R' subtype (all BCEs assigned to this port are of subtype
          '-R').
          The value of the 'office' indicates that the port is
          designated as '-O' subtype (all BCEs assigned to this port are
          of subtype '-O').
          The value of 'unknown' indicates that the port has no assigned
          BCEs yet or that the assigned BCEs are not of the same side
          (subTypeBCEMismatch).

          This object maps to the WT-159 attribute aGroupEnd."
        REFERENCE
           "[WT-159] 5.5.1.11"
        ::= { gBondPortStatusEntry 6 }

      gBondNumBCEs  OBJECT-TYPE
        SYNTAX      Unsigned32 (0..32)
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
          "Number of BCEs that is currently aggregated by the local GBS
          (assigned to the G.Bond port using ifStackTable).
          This number is never greater than gBondCapacity.

          This object SHALL be automatically incremented or decremented
          when a BCE is added or deleted to/from the G.Bond port using
          ifStackTable.

          This object maps to the WT-159 attribute aGroupNumChannels"
        REFERENCE
          "[WT-159] 5.5.1.14"
        ::= { gBondPortStatusEntry 7 }

      -- The BCE group

      gBondBceConfTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF GBondBceConfEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "Table for Configuration of G.Bond common aspects for the
          Bonding Channel Entity (BCE) ports (modems/channels).
```

```
          Entries in this table MUST be maintained in a persistent
          manner."
        ::= { gBondBce 1 }

      gBondBceConfEntry OBJECT-TYPE
        SYNTAX      GBondBceConfEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "An entry in the G.Bond BCE Configuration table.
          Each entry represents common aspects of a G.Bond BCE port
          indexed by the ifIndex. Note that an G.Bond BCE port can be
          stacked below a single GBS port, also indexed by ifIndex,
          possibly together with other BCE ports if GAF is enabled."
        INDEX  { ifIndex }
        ::= { gBondBceConfTable 1 }

      GBondBceConfEntry ::=
        SEQUENCE {
          gBondBceRemoteDiscoveryCode    PhysAddress
        }

      gBondBceRemoteDiscoveryCode  OBJECT-TYPE
        SYNTAX      PhysAddress (SIZE(0|6))
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
          "A Remote Discovery Code of the BCE port at CO.
          A 6 octet long Discovery Code of the peer GBS connected via
          the BCE.
          Reading this object results in a Discovery Get operation.
          Setting this object to all zeroes results in a Discovery
          Clear_if_Same operation (the value of gBondDiscoveryCode
          at the peer GBS SHALL be the same as gBondDiscoveryCode of
          the local GBS associated with the BCE for the operation to
          succeed).
          Writing a non-zero value to this object results in a
          Discovery Set_if_Clear operation.
          A zero-length octet string SHALL be returned on an attempt to
          read this object when GAF aggregation is not enabled.

          This object is irrelevant in BCE-R port subtypes (CPE side):
          in this case a zero length octet string SHALL be returned on
          an attempt to read this object, writing to this object SHALL
          be rejected.

          Discovery MUST be performed when the link is Down.
          Attempts to change this object MUST be rejected (in case of
```

```
          SNMP with the error inconsistentValue), if the link is Up or
          Initializing.

          This object maps to the WT-159 attribute
          aLineRemoteDiscoveryCode."
        REFERENCE
          "[WT-159] 5.5.6.7"
        ::= { gBondBceConfEntry 1 }

      --
      -- Conformance Statements
      --

       gBondGroups      OBJECT IDENTIFIER ::= { gBondConformance 1 }

       gBondCompliances OBJECT IDENTIFIER ::= { gBondConformance 2 }

       -- Object Groups

       gBondBasicGroup OBJECT-GROUP
         OBJECTS {
           gBondOperScheme,
           gBondUpDataRate,
           gBondDownDataRate,
           gBondTargetUpDataRate,
           gBondTargetDownDataRate,
           gBondOperScheme,
           gBondCapacity,
           gBondNumBCEs,
           gBondPortSide,
           gBondFltStatus
         }
         STATUS      current
         DESCRIPTION
           "A collection of objects representing management information
           common to all types of G.Bond ports."
         ::= { gBondGroups 1 }

       gBondDiscoveryGroup OBJECT-GROUP
         OBJECTS {
           gBondPeerOperScheme,
           gBondPeerCapacity,
           gBondDiscoveryCode,
           gBondBceRemoteDiscoveryCode
         }
         STATUS      current
         DESCRIPTION
           "A collection of objects supporting OPTIONAL G.Bond discovery
```

```
          in G.Bond ports."
        ::= { gBondGroups 2 }

      gBondMultiSchemeGroup OBJECT-GROUP
        OBJECTS {
          gBondSchemeSupported,
          gBondPeerSchemeSupported,
          gBondAdminScheme,
          gBondPeerAdminScheme
        }
        STATUS      current
        DESCRIPTION
          "A collection of objects providing OPTIONAL management
          information for G.Bond ports supporting multiple bonding
          schemes."
        ::= { gBondGroups 3 }

      gBondAlarmConfGroup OBJECT-GROUP
        OBJECTS {
          gBondThreshLowUpRate,
          gBondThreshLowDownRate,
          gBondLowRateCrossingEnable
        }
        STATUS      current
        DESCRIPTION
          "A collection of objects required for configuration of alarm
          thresholds and notifications in G.Bond ports."
        ::= { gBondGroups 4 }

      gBondNotificationGroup NOTIFICATION-GROUP
        NOTIFICATIONS {
          gBondLowUpRateCrossing,
          gBondLowDownRateCrossing
        }
        STATUS      current
        DESCRIPTION
          "This group supports notifications of significant conditions
          associated with G.Bond ports."
        ::= { gBondGroups 5 }

     -- Compliance Statements

      gBondCompliance MODULE-COMPLIANCE
        STATUS      current
        DESCRIPTION
          "The compliance statement for G.Bond interfaces.
          Compliance with the following external compliance statements
          is REQUIRED:
```

```
          MIB Module             Compliance Statement
          ----------             --------------------
          IF-MIB                 ifCompliance3

          Compliance with the following external compliance statements
          is OPTIONAL for implementations supporting bonding with
          flexible cross-connect between the GBS and BCE ports:

          MIB Module             Compliance Statement
          ----------             --------------------
          IF-INVERTED-STACK-MIB  ifInvCompliance
          IF-CAP-STACK-MIB       ifCapStackCompliance"

        MODULE  -- this module
          MANDATORY-GROUPS {
            gBondBasicGroup,
            gBondAlarmConfGroup,
            gBondNotificationGroup
          }

          GROUP       gBondDiscoveryGroup
          DESCRIPTION
            "Support for this group is only required for implementations
            supporting G.Bond Discovery function."

          GROUP       gBondMultiSchemeGroup
          DESCRIPTION
            "Support for this group is only required for implementations
            supporting multiple bonding schemes."

        ::= { gBondCompliances 1 }
   END
```

7. Security Considerations

There is a number of managed objects defined in the GBOND-MIB module that have a MAX-ACCESS clause of read-write or read-create. Most objects are writeable only when the link is Down. Writing to these objects can have potentially disruptive effects on network operation, for example:

- Changing of gBondAdminScheme MAY lead to a potential locking of the link, if the peer device does not support desired bonding scheme.

- Changing of gBondDiscoveryCode, before the discovery operation, MAY lead to a wrongful discovery, for example when two CO ports are connected to the same multi-channel RT port, while both CO

ports have the same Discovery register value.

- Changing GBS configuration parameters (e.g., profile of a GBS via gBondAdminProfile) MAY lead to anything from link quality and rate degradation to a complete link initialization failure, as ability of an G.Bond port to support a particular configuration depends on the copper environment.

- Activation of a specific line/channel MAY cause a severe degradation of service for another G.Bond port, whose channel(s) MAY be affected by the cross-talk from the newly activated channel.

- Removal of a channel from an operationally 'up' G.Bond port, aggregating several channels, MAY cause port's rate degradation

The user of the GBOND-MIB module must therefore be aware that support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

The readable objects in the GBOND-MIB module (i.e., those with MAX-ACCESS other than not-accessible) may be considered sensitive in some environments since, collectively, they provide information about the performance of network interfaces and can reveal some aspects of their configuration. In particular, since a bonded xDSL port can be comprised of multiple Unshielded Twisted Pair (UTP) voice grade copper, located in the same bundle with other pairs belonging to another operator/customer, it is theoretically possible to evasdrop to a G.Bond transmission, simply by "listening" to a cross-talk from the bonded pairs, especially if the parameters of the G.Bond link in question are known.

In such environments it is important to control also GET and NOTIFY access to these objects and possibly even to encrypt their values when sending them over the network via SNMP.

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

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

Further, deployment of SNMP versions prior to SNMPv3 is NOT

RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

8. IANA Considerations

Three new values of IANAifType: g9981, g9982 and g9982 SHALL be defined by the IANA [1] in the IANAifType-MIB module [IANAifType-MIB], before this document is published as an RFC.

Additionally, an object identifier for gBondMIB MODULE-IDENTITY SHALL be allocated by IANA in the MIB-2 transmission sub-tree, before this document is published.

9. Acknowledgments

This document was produced by the [ADSLMIB] working group.

10. References

10.1. Normative References

[RFC2119] 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.

[WT-159] Morgenstern, M., Beili, E., and N. Nair, "Management Framework for xDSL Bonding", DSL Forum technical report WT-159, August 2008.

10.2. Informative References

[802.3] IEEE, "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", IEEE Std 802.3-2005, December 2005.

[ADSLMIB] IETF, "ADSL MIB (adslmib) Charter", <http://www.ietf.org/html.charters/adslmib-charter.html>.

[G.991.2] ITU-T, "Single-pair High-speed Digital Subscriber Line (SHDSL) transceivers", ITU-T Recommendation G.991.2, December 2003, <http://www.itu.int/rec/T-REC-G.991.2/en>.

[G.993.1] ITU-T, "Very High speed Digital Subscriber Line transceivers", ITU-T Recommendation G.993.1, June 2004, <http://www.itu.int/rec/T-REC-G.993.1/en>.

[G.998.1] ITU-T, "ATM-based multi-pair bonding", ITU-T Recommendation G.998.1, January 2005, <http://www.itu.int/rec/T-REC-G.998.1/en>.

[G.998.2] ITU-T, "Ethernet-based multi-pair bonding", ITU-T Recommendation G.998.2,

January 2005, <http://www.itu.int/rec/T-REC-G.998.2/en>.

[G.998.3] ITU-T, "Multi-pair bonding using time-division inverse multiplexing", ITU-T Recommendation G.998.3, January 2005, <http://www.itu.int/rec/T-REC-G.998.3/en>.

[I-D.ietf-adslmib-gbond-atm-mib] Beili, E., Morgenstern, M., and N. Nair, "ATM-based xDSL Bonded Interfaces MIB", draft-ietf-adslmib-gbond-atm-mib-02 (work in progress), September 2008.

[I-D.ietf-adslmib-gbond-eth-mib] Beili, E. and M. Morgenstern, "Ethernet-based xDSL Bonded Interfaces MIB", draft-ietf-adslmib-gbond-eth-mib-02 (work in progress), September 2008.

[I-D.ietf-adslmib-gbond-tdim-mib] Beili, E. and N. Nair, "TDIM-based xDSL Bonded Interfaces MIB", draft-ietf-adslmib-gbond-tdim-mib-02 (work in progress), September 2008.

[I-D.ietf-adslmib-vdsl2-mib] Morgenstern, M., Baillie, S., and U. Bonollo, "Definitions of Managed Objects for Very High Speed Digital Subscriber Line 2 (VDSL2)", draft-ietf-adslmib-vdsl2-06 (work in progress), July 2008.

[IANAifType-MIB] Internet Assigned Numbers Authority (IANA), "IANAifType Textual Convention definition", <http://www.iana.org/assignments/ianaiftype-mib>.

[RFC2864] McCloghrie, K. and G. Hanson, "The Inverted Stack Table Extension to the Interfaces Group MIB", RFC 2864, June 2000.

[RFC3410] Case, J., Mundy, R., Partain, D.,

and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3440] Ly, F. and G. Bathrick, "Definitions of Extension Managed Objects for Asymmetric Digital Subscriber Lines", RFC 3440, December 2002.

[RFC3728] Ray, B. and R. Abbi, "Definitions of Managed Objects for Very High Speed Digital Subscriber Lines (VDSL)", RFC 3728, February 2004.

[RFC4181] Heard, C., "Guidelines for Authors and Reviewers of MIB Documents", BCP 111, RFC 4181, September 2005.

[RFC4319] Sikes, C., Ray, B., and R. Abbi, "Definitions of Managed Objects for High Bit-Rate DSL - 2nd generation (HDSL2) and Single-Pair High-Speed Digital Subscriber Line (SHDSL) Lines", RFC 4319, December 2005.

[RFC4706] Morgenstern, M., Dodge, M., Baillie, S., and U. Bonollo, "Definitions of Managed Objects for Asymmetric Digital Subscriber Line 2 (ADSL2)", RFC 4706, November 2006.

[RFC4836] Beili, E., "Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)", RFC 4836, April 2007.

[RFC5066] Beili, E., "Ethernet in the First Mile Copper (EFMCu) Interfaces MIB", RFC 5066, November 2007.

[af-phy-0086] ATM Forum, "Inverse Multiplexing for ATM (IMA) Specification Version 1.1", ATM Forum

specification af-pfy-0086.001,
March 1999.

URIs

[1] <http://www.iana.org/>

Authors' Addresses

Edward Beili
Actelis Networks
25 Bazel St.
Petach-Tikva 49103
Israel

Phone: +972-3-924-3491
EMail: edward.beili@actelis.com

Moti Morgenstern
ECI Telecom
30 Hasivim St.
Petach-Tikva 49517
Israel

Phone: +972-3-926-6258
EMail: moti.morgenstern@ecitele.com

Narendranath Nair
Wipro Technologies
Keonics Electronics City
Bangalore 560 100
India

Phone: +91-80-2852-0408 x85338
EMail: narendranath.nair@wipro.com

Full Copyright Statement

Intellectual Property