          Remote Network Monitoring MIB Extensions for Switched Networks
                              Version 1.0

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

Abstract

   This memo defines a portion of the Management Information Base (MIB)
   for use with network management protocols in TCP/IP-based internets.
   In particular, it defines objects for managing remote network
   monitoring devices in switched networks environments.

Table of Contents

1. The Network Management Framework

   The SNMP Management Framework presently consists of five major
   components:

   - An overall architecture, described in RFC 2571 [1].

   - Mechanisms for describing and naming objects and events for the
     purpose of management. The first version of this Structure of
     Management Information (SMI) is called SMIv1 and described in STD
     16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second
     version, called SMIv2, is described in STD 58, RFC 2578 [5], RFC
     2579 [6] and RFC 2580 [7].

   - Message protocols for transferring management information. The
     first version of the SNMP message protocol is called SNMPv1 and
     described in STD 15, RFC 1157 [8]. A second version of the SNMP
     message protocol, which is not an Internet standards track
     protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC
     1906 [10]. The third version of the message protocol is called
     SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574
     [12].

   - Protocol operations for accessing management information. The first
     set of protocol operations and associated PDU formats is described
     in STD 15, RFC 1157 [8]. A second set of protocol operations and
     associated PDU formats is described in RFC 1905 [13].

   - A set of fundamental applications described in RFC 2573 [14] and
     the view-based access control mechanism described in RFC 2575 [15].

Managed objects are accessed via a virtual information store, termed
the Management Information Base or MIB. Objects in the MIB are
defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A
MIB conforming to the SMIv1 can be produced through the appropriate
translations. The resulting translated MIB must be semantically
equivalent, except where objects or events are omitted because no
information in SMIv2 will be converted into textual descriptions in
SMIv1 during the translation process. However, this loss of machine
readable information is not considered to change the semantics of the
MIB.

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 [24].

2. Overview

This document continues the architecture created in the RMON MIB [17]
by providing RMON analysis for switched networks (SMON).

Remote network monitoring devices, often called monitors or probes,
are instruments that exist for the purpose of managing a network.
Often these remote probes are stand-alone devices and devote
significant internal resources for the sole purpose of managing a
network. An organization may employ many of these devices, one per
network segment, to manage its internet. In addition, these devices
may be used for a network management service provider to access a
client network, often geographically remote.

The objects defined in this document are intended as an interface
between an RMON agent and an RMON management application and are not
intended for direct manipulation by humans. While some users may
tolerate the direct display of some of these objects, few will
tolerate the complexity of manually manipulating objects to
accomplish row creation. These functions should be handled by the
management application.

2.1 Remote Network Management Goals

   o Offline Operation

     There are sometimes conditions when a management station will not
     be in constant contact with its remote monitoring devices. This is
     sometimes by design in an attempt to lower communications costs

(especially when communicating over a WAN or dialup link), or by
accident as network failures affect the communications between the
management station and the probe.

For this reason, this MIB allows a probe to be configured to
perform diagnostics and to collect statistics continuously, even
when communication with the management station may not be possible
or efficient. The probe may then attempt to notify the management
station when an exceptional condition occurs. Thus, even in
circumstances where communication between management station and
probe is not continuous, fault, performance, and configuration
information may be continuously accumulated and communicated to the
management station conveniently and efficiently.

o Proactive Monitoring

  Given the resources available on the monitor, it is potentially
  helpful for it continuously to run diagnostics and to log network
  performance. The monitor is always available at the onset of any
  failure. It can notify the management station of the failure and
  can store historical statistical information about the failure.
  This historical information can be played back by the management
  station in an attempt to perform further diagnosis into the cause
  of the problem.

o Problem Detection and Reporting

  The monitor can be configured to recognize conditions, most notably
  error conditions, and continuously to check for them. When one of
  these conditions occurs, the event may be logged, and management
  stations may be notified in a number of ways.

o Value Added Data

  Because a remote monitoring device represents a network resource
  dedicated exclusively to network management functions, and because
  it is located directly on the monitored portion of the network, the
  remote network monitoring device has the opportunity to add
  significant value to the data it collects. For instance, by
  highlighting those hosts on the network that generate the most
  traffic or errors, the probe can give the management station
  precisely the information it needs to solve a class of problems.

o Multiple Managers

  An organization may have multiple management stations for different
  units of the organization, for different functions (e.g.
  engineering and operations), and in an attempt to provide disaster

recovery. Because environments with multiple management stations
are common, the remote network monitoring device has to deal with
more than one management station, potentially using its resources
concurrently.

2.2 Switched Networks Monitoring

This document addresses issues related to applying "Remote
Technology" to Switch Networks. Switches today differ from standard
shared media protocols:

1)   Data is not, in general, broadcast. This MAY be caused by the
     switch architecture or by the connection-oriented nature of the
     data. This means, therefore, that monitoring non-broadcast
     traffic needs to be considered.

2)   Monitoring the multiple entry and exit points from a switching
     device requires a vast amount of resources - memory and CPU, and
     aggregation of the data in logical packets of information,
     determined by the application needs.

3)   Switching incorporates logical segmentation such as Virtual LANs
     (VLANs).

4)   Switching incorporates packet prioritization.

5)   Data across the switch fabric can be in the form of cells. Like
     RMON, SMON is only concerned with the monitoring of packets.

Differences such as these make monitoring difficult. The current
RMON and RMON 2 standards do not provide for things that are unique
to switches or switched environments.

In order to overcome the limitations of the existing standards, new
monitoring mechanisms have been implemented by vendors of switching
equipment. All these monitoring strategies are currently proprietary
in nature.

This document provides the framework to include different switching
strategies and allow for monitoring operations consistent with the
RMON framework. This MIB is limited to monitoring and control
operations aimed at providing monitoring data for RMON probes.

2.3 Mechanisms for Monitoring Switched Networks

The following mechanisms are used by SMON devices, for the purpose of
monitoring switched networks.

2.3.1 DataSource Objects

   The RMON MIB standard [17] defines data source objects which point to
   MIB-II interfaces, identified by instances of ifIndex objects.

   The SMON MIB extends this concept and allows for other types of
   objects to be defined as data sources for RMON and/or SMON data.
   Three forms of dataSources are described:

      ifIndex.<I>

         Traditional RMON dataSources. Called 'port-based' for
         ifType.<I> not equal to 'propVirtual(53)'. <I> is the ifIndex
         value (see [22]).

      smonVlanDataSource.<V>

         A dataSource of this form refers to a 'Packet-based VLAN' and
         is called a 'VLAN-based' dataSource. <V> is the VLAN ID as
         defined by the IEEE 802.1Q standard [19]. The value is between
         1 and 4094 inclusive, and it represents an 802.1Q VLAN-ID with
         global scope within a given bridged domain, as defined by [19].

      entPhysicalEntry.<N>

         A dataSource of this form refers to a physical entity within
         the agent and is called an 'entity-based' dataSource. <N> is
         the value of the entPhysicalIndex in the entPhysicalTable (see
         [18]).

   In addition to these new dataSource types, SMON introduces a new
   group called dataSourceCapsTable to aid an NMS in discovering
   dataSource identity and attributes.

   The extended data source mechanism supported by the SMON MIB allows
   for the use of external collection points, similar to the one defined
   and supported by the RMON and RMON 2 MIBs, as well as internal
   collection points (e.g. propVirtual ifTable entry, entPhysicalEntry).
   The latter reflects either data sources which MAY be the result of
   aggregation (e.g. switch-wide) or internal channels of physical
   entities, which have the capability of being monitored by an SMON
   probe.

2.3.2 Copy Port

   In order to make the switching devices support RMON statistics, many
   vendors have implemented a port copy feature, allowing traffic to be
   replicated from switch port to switch port. Several levels of
   configuration are possible:

      1) 1 source port to 1 destination port
      2) N source ports to 1 destination port
      3) N source ports to M destination ports

   The SMON standard presents a standard MIB interface which allows for
   the control of this function.

   Note that this function can apply to devices that have no other SMON
   or RMON functionality than copy port. The agent of such a device
   would support only the portCopyCaps and the portCopyConfig MIB
   groups, out of the whole SMON MIB. Switch vendors are encouraged to
   implement this subset of the SMON MIB, as it would allow for standard
   port copy configuration from the same NMS application that does RMON
   or SMON.

   Port copy may cause congestion problems on the SMON device. This
   situation is more likely occur when copying from a port of higher
   speed to a port of lower speed or copy from multiple port to a single
   port.

   Particular implementations MAY chose to build protection mechanisms
   that would prevent creation of new port copy links when the capacity
   of the destination port is exceeded. The MIB allows for
   implementations to (if supported) instrument a destination drop count
   on port copy to provide NMS applications a sense of the quality of
   data presented at the destination port.

2.3.3 VLAN Monitoring

   VLAN monitoring can be accomplished by using a VLAN-based dataSource
   and/or by configuring smonVlanIdStats and/or smonPrioStats
   collections. These functions allow VLAN-ID or user priority
   distributions per dataSource. VLAN monitoring provides a high-level
   view of total VLAN usages and relative non-unicast traffic usage as
   well as a profile of VLAN priority as defined in the 3-bit
   user_priority field.

   NOTE: priority statistics reflect what was parsed from the packet,
   not what priority, if any, was necessarily granted by the switch.

2.4 Relationship to Other MIBs

2.4.1 The RMON and RMON 2 MIBs

   The Remote Monitoring MIB (RMON) [17] provides several management
   functions that MAY be directly or indirectly applicable to switched
   networks.

   The port copy mechanisms defined by the SMON MIB allow for the
   destination ports to become a data source for any RMON statistics.
   However, an NMS application SHOULD check whether it is in the device
   capability (portCopyCap) to filter errors from a source to a
   destination port and whether this capability is enabled, in order to
   provide a correct interpretation of the copied port traffic.

   RMON host and matrix group statistics entries MAY be aggregated by
   use of the extended dataSource capability defined in SMON. RMON 2
   groups are similarly extended through the use of SMON's dataSource
   definition.

   RMON also defines a simple thresholding monitoring mechanism, event-
   logging and event-notification for any MIB instance; SMON utilizes
   the alarms and events groups from RMON without modification. These
   groups SHOULD be implemented on SMON devices if a simple thresholding
   mechanism is desired.

   The RMON 2 usrHistory group (user-defined history collection) SHOULD
   be implemented by an SMON device if a history collection mechanism is
   desired for smonStats entries.

2.4.2 The Interfaces Group MIB

   The SMON MIB utilizes the propVirtual(53) ifType defined in the
   Interfaces Group MIB [22] to provide SMON and RMON with new
   dataSources such as VLANs and internal monitoring points. NMS
   applications SHOULD consult the SMON dataSource capabilities group
   (dataSourceCap) for a description of these virtual interfaces.

2.4.3 The Entity MIB

   The SMON MIB does not mandate Entity MIB [18] support, but allows for
   physical entities, as defined by this MIB to be defined as SMON data
   sources. For such cases, the support for the entPhysicalTable is
   required.

2.4.4 The Bridge MIB

   One of the important indicators for measuring the effectiveness of a
   switching device is the ratio between the number of forwarded frames
   and the number of dropped frames at the switch port.

   It is out of the scope of this MIB to provide instrumentation
   information relative to switching devices. However, such indication
   may be part of other MIB modules.

   For instance the Bridge MIB [23] provides such MIB objects, for the
   802.1 bridges (dot1dTpPortInFrames, dot1dTpPortInDiscards) and
   switches managed according to the 802.1 bridge model MAY provide this
   information.

2.5 Relationship with IEEE 802.1 Standards

   The SMON MIB provides simple statistics per VLAN and priority levels.
   Those two categories of statistics are important to managers of
   switched networks. Interoperability for those features is ensured by
   the use of the IEEE 802.1 p/Q standards ([19], [20]) defined by the
   IEEE 802.1 WG. Interoperability from the SMON MIB point of view is
   ensured by referencing the IEEE definition of VLANs and priority
   levels for the SMON statistics.

3. SMON Groups

3.1 SMON ProbeCapabilities

   The SMON probeCapabilities BITS object covers the following four
   capabilities.

      - smonVlanStats(0)
        The probe supports the smonVlanStats object group.

      - smonPrioStats(1)
        The probe supports the smonPrioStats object group.

      - dataSource(2)
        The probe supports the dataSourceCaps object group.

      - portCopy(4)
        The probe supports the portCopyConfig object group.

3.2 smonVlanStats

The smonVlanStats MIB group includes the control and statistics
objects related to 802.1Q VLANs. Specific statistics per 802.1Q
virtual LAN are supported. The group provides a high level view of
total VLAN usage, and relative non-unicast traffic usage.

It is an implementation-specific matter as to how the agent
determines the proper default-VLAN for untagged or priority-tagged
frames.

3.3 smonPrioStats

The smonPrioStatsTable provides a distribution based on the
user_priority field in the VLAN header.

Note that this table merely reports priority as encoded in VLAN
headers, not the priority (if any) given the frame for actual
switching purposes.

3.4 dataSourceCaps

The dataSourceCaps MIB group identifies all supported data sources on
an SMON device. An NMS MAY use this table to discover the RMON and
Copy Port attributes of each data source.

Upon restart of the agent, the dataSourceTable, ifTable and
entPhysicalTable are initialized for the available data sources. The
agent MAY modify these tables as data sources become known or are
removed (e.g. hot swap of interfaces, chassis cards or the discovery
of VLAN usage). It is understood that dataSources representing VLANs
may not always be instantiated immediately upon restart, but rather
as VLAN usage is detected by the agent. The agent SHOULD attempt to
create dataSource and interface entries for all dataSources as soon
as possible.

For each dataSourceCapsEntry representing a VLAN or entPhysicalEntry,
the agent MUST create an associated ifEntry with a ifType value of
associated dataSourceCapsIfIndex object.

The rationale of the above derives from the fact that according to
[16] and [17] an RMON dataSource MUST be associated with an ifEntry.
Specifically, the dataSourceCapsTable allows for an agent to map
Entity MIB physical entities (e.g., switch backplanes) and entire
VLANs to ifEntries with ifType "propVirtual(53)". This ifEntry values
will be used as the actual values in RMON control table dataSource
objects. This allows for physical entities and VLANs to be treated
as RMON data sources, and RMON functions to be applied to this type

of data sources.

3.5 portCopyConfig

   The portCopyConfig MIB group includes the objects defined for the
   control of the port copy functionality in a device.

   The standard does not place a limit on the mode in which this copy
   function may be used:

   Mode 1 --   1:1 Copy

      Single dataSource copied to a single destination dataSource.
      Agent MAY limit configuration based on ifTypes, ifSpeeds, half-
      duplex/full-duplex, or agent resources. In this mode the single
      instance of the portCopyDestDropEvents object refers to dropped
      frames on the portCopyDest interface.

   Mode 2 --   N:1 Copy

      Multiple dataSources copied to a single destination dataSource.
      Agent MAY limit configuration based on ifTypes, ifSpeeds, half-
      duplex/full-duplex, portCopyDest over-subscription, or agent
      resources. In this mode all N instances of the
      portCopyDestDropEvents object SHOULD contain the same value, and
      refer to dropped frames on the portCopyDest interface.

   Mode 3 --   N:M Copy

      Multiple dataSources copied to multiple destination dataSources.
      Agent MAY limit configuration based on ifTypes, ifSpeeds, half-
      duplex/full-duplex, portCopyDest over-subscription, or agent
      resources. Since portCopyDestDropEvents is kept per destination
      port, all instances of the portCopyDestDropEvents object
      associated with (indexed by) a given portCopyDest SHOULD have the
      same value (i.e. replicated or aliased for each instance
      associated with a given portCopyDest).

   The rows do not have an OwnerString, since multiple rows MAY be part
   of the same portCopy operation. The agent is expected to activate or
   deactivate entries one at a time, based on the rowStatus for the
   given row. This can lead to unpredictable results in Modes 2 and 3
   in applications utilizing the portCopy target traffic, if multiple
   PDUs are used to fully configure the operation. It is RECOMMENDED
   that an entire portCopy operation be configured in one SetRequest PDU
   if possible.

The portCopyDest object MAY NOT reference an interface associated
with a packet-based VLAN (smonVlanDataSource.<V>), but this
dataSource type MAY be used as a portCopySource.

4. Control of Remote Network Monitoring Devices

Due to the complex nature of the available functions in these
devices, the functions often need user configuration. In many cases,
the function requires parameters to be set up for a data collection
operation. The operation can proceed only after these parameters are
fully set up.

Many functional groups in this MIB have one or more tables in which
to set up control parameters, and one or more data tables in which to
place the results of the operation. The control tables are typically
read/write in nature, while the data tables are typically read-only.
Because the parameters in the control table often describe resulting
data in the data table, many of the parameters can be modified only
when the control entry is not active. Thus, the method for modifying
these parameters is to de-activate the entry, perform the SNMP Set
operations to modify the entry, and then re-activate the entry.
Deleting the control entry causes the deletion of any associated data
entries, which also gives a convenient method for reclaiming the
resources used by the associated data.

Some objects in this MIB provide a mechanism to execute an action on
the remote monitoring device. These objects MAY execute an action as
a result of a change in the state of the object. For those objects
in this MIB, a request to set an object to the same value as it
currently holds would thus cause no action to occur.

To facilitate control by multiple managers, resources have to be
shared among the managers. These resources are typically the memory
and computation resources that a function requires.

The control mechanisms defined and used in this MIB are the same as
those defined in the RMON MIB [17], for control functionality and
interaction with multiple managers.

5. Definitions

    SMON-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE, Counter32,
        Integer32, Counter64
                FROM SNMPv2-SMI
        RowStatus, TEXTUAL-CONVENTION
                FROM SNMPv2-TC
        rmon, OwnerString
                FROM RMON-MIB
        LastCreateTime, DataSource, rmonConformance, probeConfig
                FROM RMON2-MIB
        InterfaceIndex
                FROM IF-MIB
        MODULE-COMPLIANCE, OBJECT-GROUP
                FROM SNMPv2-CONF;

    switchRMON MODULE-IDENTITY
        LAST-UPDATED "9812160000Z"
        ORGANIZATION "IETF RMON MIB Working Group"
        CONTACT-INFO
        "IETF RMONMIB WG Mailing list: rmonmib@cisco.com

                Rich Waterman
                Allot Networks Inc.
                Tel: +1-408-559-0253
                Email: rich@allot.com

                Bill Lahaye
                Xylan Corp.
                Tel: +1-800-995-2612
                Email: lahaye@ctron.com

                Dan Romascanu
                Lucent Technologies
                Tel: +972-3-645-8414
                Email: dromasca@lucent.com

                Steven Waldbusser
                International Network Services (INS)
                Tel: +1-650-318-1251
                Email: waldbusser@ins.com"

        DESCRIPTION
                "The MIB module for managing remote monitoring device
                 implementations for Switched Networks"

```
        -- revision history

        REVISION    "9812160000Z"     -- 16 Dec 1998 midemight
        DESCRIPTION "Initial Version, published as RFC 2613."

        ::= { rmon 22 }

 smonMIBObjects OBJECT IDENTIFIER ::= { switchRMON 1 }

 dataSourceCaps          OBJECT IDENTIFIER ::= {smonMIBObjects 1}
 smonStats               OBJECT IDENTIFIER ::= {smonMIBObjects 2}
 portCopyConfig          OBJECT IDENTIFIER ::= {smonMIBObjects 3}
 smonRegistrationPoints  OBJECT IDENTIFIER ::= {smonMIBObjects 4}

 -- Textual Conventions
 --

 SmonDataSource ::= TEXTUAL-CONVENTION
  STATUS       current
  DESCRIPTION
      "Identifies the source of the data that the associated function
       is configured to analyze. This Textual Convention
       extends the DataSource Textual Convention defined by RMON 2
       to the following data source types:

       - ifIndex.<I>
       DataSources of this traditional form are called 'port-based',
       but only if ifType.<I> is not equal to 'propVirtual(53)'.

       - smonVlanDataSource.<V>
       A dataSource of this form refers to a 'Packet-based VLAN'
       and is called a 'VLAN-based' dataSource. <V> is the VLAN
       ID as defined by the IEEE 802.1Q standard [19]. The
       value is between 1 and 4094 inclusive, and it represents
       an 802.1Q VLAN-ID with global scope within a given
       bridged domain, as defined by [19].

      - entPhysicalEntry.<N>
       A dataSource of this form refers to a physical entity within
       the agent (e.g. entPhysicalClass = backplane(4)) and is called
       an 'entity-based' dataSource."
  SYNTAX       OBJECT IDENTIFIER

 -- The smonCapabilities object describes SMON agent capabilities.

 smonCapabilities OBJECT-TYPE
  SYNTAX BITS {
    smonVlanStats(0),
```

```
      smonPrioStats(1),
      dataSource(2),
      smonUnusedBit(3),
      portCopy(4)
      }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
      "An indication of the SMON MIB groups supported
       by this agent."
  ::= { probeConfig  15 }

  -- dataSourceCaps MIB group - defines SMON data source and port
  -- copy capabilities for devices supporting SMON.

  -- A NMS application will check this MIB group and retrieve
  -- information about the SMON capabilities of the device before
  -- applying SMON control operations to the device.


  -- dataSourceCapsTable: defines capabilities of RMON data sources

  dataSourceCapsTable OBJECT-TYPE
   SYNTAX       SEQUENCE OF DataSourceCapsEntry
   MAX-ACCESS  not-accessible
   STATUS       current
   DESCRIPTION
      "This table describes RMON data sources and port copy
      capabilities. An NMS MAY use this table to discover the
      identity and attributes of the data sources on a given agent
      implementation. Similar to the probeCapabilities object,
      actual row-creation operations will succeed or fail based on
      the resources available and parameter values used in each
      row-creation operation.

      Upon restart of the RMON agent, the dataSourceTable, ifTable,
      and perhaps entPhysicalTable are initialized for the available
      dataSources.

      For each dataSourceCapsEntry representing a VLAN or
      entPhysicalEntry the agent MUST create an associated ifEntry
      with a ifType value of 'propVirtual(53)'. This ifEntry will be
      used as the actual value in RMON control table dataSource
      objects. The assigned ifIndex value is copied into the
      associated dataSourceCapsIfIndex object.

      It is understood that dataSources representing VLANs may not
      always be instantiated immediately upon restart, but rather as
```

```
      VLAN usage is detected by the agent. The agent SHOULD attempt
      to create dataSource and interface entries for all dataSources
      as soon as possible."
 ::= { dataSourceCaps 1 }

dataSourceCapsEntry  OBJECT-TYPE
 SYNTAX       DataSourceCapsEntry
 MAX-ACCESS   not-accessible
 STATUS       current
 DESCRIPTION
     "Entries per data source containing descriptions of data
      source and port copy capabilities. This table is populated by
      the SMON agent with one entry for each supported data
      source."
 INDEX { IMPLIED dataSourceCapsObject }
 ::= { dataSourceCapsTable 1 }

DataSourceCapsEntry ::= SEQUENCE {
     dataSourceCapsObject
             SmonDataSource,
     dataSourceRmonCaps
             BITS,
     dataSourceCopyCaps
             BITS,
     dataSourceCapsIfIndex
             InterfaceIndex
        }

dataSourceCapsObject OBJECT-TYPE
 SYNTAX       SmonDataSource
 MAX-ACCESS   not-accessible
 STATUS       current
 DESCRIPTION
     "Defines an object that can be a SMON data source or a
       source or a destination for a port copy operation."
  ::= { dataSourceCapsEntry 1  }

dataSourceRmonCaps OBJECT-TYPE
 SYNTAX BITS {
        countErrFrames(0),
        countAllGoodFrames(1),
        countAnyRmonTables(2),
        babyGiantsCountAsGood(3)
        }
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
```

          " General attributes of the specified dataSource. Note that
          these are static attributes, which SHOULD NOT be adjusted
          because of current resources or configuration.

        - countErrFrames(0)
          The agent sets this bit for the dataSource if errored frames
          received on this dataSource can actually be monitored by the
          agent The agent clears this bit if any errored frames are
          not visible to the RMON data collector.

        - countAllGoodFrames(1)
          The agent sets this bit for the dataSource if all good
          frames received on this dataSource can actually be monitored
          by the agent. The agent clears this bit if any good frames
          are not visible for RMON collection, e.g., the dataSource is
          a non-promiscuous interface or an internal switch interface
          which may not receive frames which were switched in hardware
          or dropped by the bridge forwarding function.

        - countAnyRmonTables(2)
          The agent sets this bit if this dataSource can actually be
          used in any of the implemented RMON tables, resources
          notwithstanding. The agent clears this bit if this
          dataSourceCapsEntry is present simply to identify a
          dataSource that may only be used as portCopySource and/or a
          portCopyDest, but not the source of an actual RMON data
          collection.

        - babyGiantsCountAsGood(3)
          The agent sets this bit if it can distinguish, for counting
          purposes, between true giant frames and frames that exceed
          Ethernet maximum frame size 1518 due to VLAN tagging ('baby
          giants'). Specifically, this BIT means that frames up to
          1522 octets are counted as good.

          Agents not capable of detecting 'baby giants' will clear
          this bit and will view all frames less than or equal to 1518
          octets as 'good frames' and all frames larger than 1518
          octets as 'bad frames' for the purpose of counting in the
          smonVlanIdStats and smonPrioStats tables.

          Agents capable of detecting 'baby giants' SHALL consider
          them as 'good frames' for the purpose of counting in the
          smonVlanIdStats and smonPrioStats tables."

     ::= { dataSourceCapsEntry 2 }

   dataSourceCopyCaps OBJECT-TYPE

```
      SYNTAX BITS {
          copySourcePort(0),
          copyDestPort(1),
          copySrcTxTraffic(2),
          copySrcRxTraffic(3),
          countDestDropEvents(4),
          copyErrFrames(5),
          copyUnalteredFrames(6),
          copyAllGoodFrames(7)
          }
  MAX-ACCESS    read-only
  STATUS        current
  DESCRIPTION
      "PortCopy function capabilities of the specified dataSource.
  Note that these are static capabilities, which SHOULD NOT be
  adjusted because of current resources or configuration.

    - copySourcePort(0)
        The agent sets this bit if this dataSource is capable of
        acting as a source of a portCopy operation. The agent clears
        this bit otherwise.

    - copyDestPort(1)
        The agent sets this bit if this dataSource is capable of
        acting as a destination of a portCopy operation. The agent
        clears this bit otherwise.

   - copySrcTxTraffic(2)
        If the copySourcePort bit is set:
            The agent sets this bit if this dataSource is capable of
          copying frames transmitted out this portCopy source. The
          agent clears this bit otherwise. This function is needed
          to support full-duplex ports.
         Else:
            this bit SHOULD be cleared.

    - copySrcRxTraffic(3)
        If the copySourcePort bit is set:
          The agent sets this bit if this dataSource is capable of
          copying frames received on this portCopy source. The agent
          clears this bit otherwise. This function is needed to
          support full-duplex ports.
        Else:
          this bit SHOULD be cleared.

    - countDestDropEvents(4)
        If the copyDestPort bit is set:
            The agent sets this bit if it is capable of incrementing
```

                   portCopyDestDropEvents, when this dataSource is the
                   target of a portCopy operation and a frame destined to
                   this dataSource is dropped (for RMON counting purposes).
               Else:
                   this BIT SHOULD be cleared.

          - copyErrFrames(5)
               If the copySourcePort bit is set:
                   The agent sets this bit if it is capable of copying all
                   errored frames from this portCopy source-port, for
                   errored frames received on this dataSource.
               Else:
                   this BIT SHOULD be cleared.

          - copyUnalteredFrames(6)
               If the copySourcePort bit is set:
                   The agent sets the copyUnalteredFrames bit If it is
                   capable of copying all frames from this portCopy
                   source-port without alteration in any way;
               Else:
                   this bit SHOULD be cleared.

          - copyAllGoodFrames(7)
               If the copySourcePort bit is set:
                   The agent sets this bit for the dataSource if all good
                   frames received on this dataSource are normally capable
                   of being copied by the agent. The agent clears this bit
                   if any good frames are not visible for the RMON portCopy
                   operation, e.g., the dataSource is a non-promiscuous
                   interface or an internal switch interface which may not
                   receive frames which were switched in hardware or
                   dropped by the bridge forwarding function.
                Else:
                   this bit SHOULD be cleared."

     ::= { dataSourceCapsEntry 3 }

    dataSourceCapsIfIndex OBJECT-TYPE
    SYNTAX   InterfaceIndex
    MAX-ACCESS   read-only
    STATUS   current
    DESCRIPTION
         "This object contains the ifIndex value of the ifEntry
          associated with this smonDataSource. The agent MUST create
          'propVirtual' ifEntries for each dataSourceCapsEntry of type
          VLAN or entPhysicalEntry."

     ::= { dataSourceCapsEntry 4 }

    -- The SMON Statistics MIB Group

    -- aggregated statistics for IEEE 802.1Q VLAN environments.

    -- VLAN statistics can be gathered by configuring smonVlanIdStats
    -- and/or smonPrioStats collections. These functions allow a
    -- VLAN-ID or user priority distributions per dataSource,
    -- auto-populated by the agent in a manner similar to the RMON
    -- hostTable.

    -- Only good frames are counted in the tables described in this
    -- section.

    -- VLAN ID Stats

    -- smonVlanStatsControlTable allows configuration of VLAN-ID
    -- collections.

    smonVlanStatsControlTable OBJECT-TYPE
     SYNTAX   SEQUENCE OF SmonVlanStatsControlEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION

         "Controls the setup of VLAN statistics tables.

          The statistics collected represent a distribution based on
          the IEEE 802.1Q VLAN-ID (VID), for each good frame attributed
          to the data source for the collection."
     ::= { smonStats 1 }

    smonVlanStatsControlEntry OBJECT-TYPE
     SYNTAX   SmonVlanStatsControlEntry
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
         "A conceptual row in the smonVlanStatsControlTable."
     INDEX { smonVlanStatsControlIndex }
     ::= { smonVlanStatsControlTable 1 }

    SmonVlanStatsControlEntry ::= SEQUENCE {
     smonVlanStatsControlIndex   Integer32,
     smonVlanStatsControlDataSource   DataSource,
     smonVlanStatsControlCreateTime   LastCreateTime,
     smonVlanStatsControlOwner   OwnerString,
     smonVlanStatsControlStatus   RowStatus
    }

    smonVlanStatsControlIndex OBJECT-TYPE
     SYNTAX   Integer32 (1..65535)
     MAX-ACCESS   not-accessible
     STATUS   current
     DESCRIPTION
         "A unique arbitrary index for this smonVlanStatsControlEntry."
     ::= { smonVlanStatsControlEntry 1 }

    smonVlanStatsControlDataSource OBJECT-TYPE
     SYNTAX   DataSource
     MAX-ACCESS   read-create
     STATUS   current
     DESCRIPTION
         "The source of data for this set of VLAN statistics.

         This object MAY NOT be modified if the associated
         smonVlanStatsControlStatus object is equal to active(1)."
     ::= { smonVlanStatsControlEntry 2 }

    smonVlanStatsControlCreateTime OBJECT-TYPE
     SYNTAX   LastCreateTime
     MAX-ACCESS read-only
     STATUS   current
     DESCRIPTION
         "The value of sysUpTime when this control entry was last
         activated. This object allows to a management station to
         detect deletion and recreation cycles between polls."
     ::= { smonVlanStatsControlEntry 3 }

    smonVlanStatsControlOwner OBJECT-TYPE

     SYNTAX   OwnerString
     MAX-ACCESS   read-create
     STATUS   current
     DESCRIPTION
         "Administratively assigned named of the owner of this entry.
         It usually defines the entity that created this entry and is
         therefore using the resources assigned to it, though there is
         no enforcement mechanism, nor assurance that rows created are
         ever used."
     ::= { smonVlanStatsControlEntry 4 }

    smonVlanStatsControlStatus OBJECT-TYPE
     SYNTAX   RowStatus
     MAX-ACCESS   read-create
     STATUS   current
     DESCRIPTION
         "The status of this row.

         An entry MAY NOT exist in the active state unless all
         objects in the entry have an appropriate value.

         If this object is not equal to active(1), all associated
         entries in the smonVlanIdStatsTable SHALL be deleted."
    ::= { smonVlanStatsControlEntry 5 }

    -- The VLAN Statistics Table

    smonVlanIdStatsTable OBJECT-TYPE
     SYNTAX   SEQUENCE OF SmonVlanIdStatsEntry
     MAX-ACCESS not-accessible
     STATUS   current
     DESCRIPTION
         "Contains the VLAN statistics data.
          The statistics collected represent a distribution based
          on the IEEE 802.1Q VLAN-ID (VID), for each good frame
          attributed to the data source for the collection.

          This function applies the same rules for attributing frames
          to VLAN-based collections. RMON VLAN statistics are collected
          after the Ingress Rules defined in section 3.13 of the VLAN
          Specification [20] are applied.

          It is possible that entries in this table will be
          garbage-collected, based on agent resources, and VLAN
          configuration. Agents are encouraged to support all 4094
          index values and not garbage collect this table."
    ::= { smonStats 2 }

    smonVlanIdStatsEntry OBJECT-TYPE
     SYNTAX   SmonVlanIdStatsEntry
     MAX-ACCESS not-accessible
     STATUS   current
     DESCRIPTION
         "A conceptual row in smonVlanIdStatsTable."
     INDEX { smonVlanStatsControlIndex, smonVlanIdStatsId }
     ::= { smonVlanIdStatsTable 1 }

    SmonVlanIdStatsEntry ::= SEQUENCE {
     smonVlanIdStatsId   Integer32,
     smonVlanIdStatsTotalPkts   Counter32,
     smonVlanIdStatsTotalOverflowPkts   Counter32,
     smonVlanIdStatsTotalHCPkts   Counter64,
     smonVlanIdStatsTotalOctets   Counter32,
     smonVlanIdStatsTotalOverflowOctets   Counter32,
     smonVlanIdStatsTotalHCOctets   Counter64,
     smonVlanIdStatsNUcastPkts   Counter32,

```
 smonVlanIdStatsNUcastOverflowPkts               Counter32,
 smonVlanIdStatsNUcastHCPkts                     Counter64,
 smonVlanIdStatsNUcastOctets                     Counter32,
 smonVlanIdStatsNUcastOverflowOctets             Counter32,
 smonVlanIdStatsNUcastHCOctets                   Counter64,
 smonVlanIdStatsCreateTime                       LastCreateTime
}

smonVlanIdStatsId OBJECT-TYPE
 SYNTAX      Integer32 (1..4094)
 MAX-ACCESS not-accessible
 STATUS      current
 DESCRIPTION
     "The unique identifier of the VLAN monitored for
      this specific statistics collection.

     Tagged packets match the VID for the range between 1 and 4094.
     An external RMON probe MAY detect VID=0 on an Inter Switch
     Link, in which case the packet belongs to a VLAN determined by
     the PVID of the ingress port. The VLAN to which such a packet
     belongs can be determined only by a RMON probe internal to the
     switch."
 REFERENCE
     "Draft Standard for Virtual Bridged Local Area Networks,
       P802.1Q/D10, chapter 3.13"
 ::= { smonVlanIdStatsEntry 1 }

smonVlanIdStatsTotalPkts OBJECT-TYPE
 SYNTAX      Counter32
 UNITS "packets"
 MAX-ACCESS read-only
 STATUS      current
 DESCRIPTION
     "The total number of packets counted on this VLAN."
 ::= { smonVlanIdStatsEntry 2 }

smonVlanIdStatsTotalOverflowPkts OBJECT-TYPE
 SYNTAX      Counter32
 UNITS "packets"
 MAX-ACCESS read-only
 STATUS      current
 DESCRIPTION
     "The number of times the associated smonVlanIdStatsTotalPkts
      counter has overflowed."
 ::= { smonVlanIdStatsEntry 3 }

smonVlanIdStatsTotalHCPkts OBJECT-TYPE
 SYNTAX      Counter64
```

```
      UNITS "packets"
      MAX-ACCESS read-only
      STATUS      current
      DESCRIPTION
          "The total number of packets counted on this VLAN."
       ::= { smonVlanIdStatsEntry 4 }

    smonVlanIdStatsTotalOctets OBJECT-TYPE
      SYNTAX      Counter32
      UNITS "octets"
      MAX-ACCESS read-only
      STATUS      current
      DESCRIPTION
          "The total number of octets counted on this VLAN."
       ::= { smonVlanIdStatsEntry 5 }

    smonVlanIdStatsTotalOverflowOctets OBJECT-TYPE
      SYNTAX      Counter32
      UNITS "octets"
      MAX-ACCESS read-only
      STATUS      current
      DESCRIPTION
          "The number of times the associated smonVlanIdStatsTotalOctets
           counter has overflowed."
       ::= { smonVlanIdStatsEntry 6 }

    smonVlanIdStatsTotalHCOctets OBJECT-TYPE
      SYNTAX      Counter64
      UNITS "octets"
      MAX-ACCESS read-only
      STATUS      current
      DESCRIPTION
          "The total number of octets counted on this VLAN."
     ::= { smonVlanIdStatsEntry 7 }

    smonVlanIdStatsNUcastPkts OBJECT-TYPE
      SYNTAX      Counter32
      UNITS "packets"
      MAX-ACCESS read-only
      STATUS      current
      DESCRIPTION
          "The total number of non-unicast packets counted on this
           VLAN."
       ::= { smonVlanIdStatsEntry 8 }

    smonVlanIdStatsNUcastOverflowPkts OBJECT-TYPE
      SYNTAX      Counter32
      UNITS "packets"
```

```
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The number of times the associated smonVlanIdStatsNUcastPkts
           counter has overflowed."
       ::= { smonVlanIdStatsEntry 9 }

      smonVlanIdStatsNUcastHCPkts OBJECT-TYPE
       SYNTAX      Counter64
       UNITS "packets"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of non-unicast packets counted on
           this VLAN."
       ::= { smonVlanIdStatsEntry 10 }

      smonVlanIdStatsNUcastOctets OBJECT-TYPE
       SYNTAX      Counter32
       UNITS "octets"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of non-unicast octets counted on
           this VLAN."
       ::= { smonVlanIdStatsEntry 11 }

      smonVlanIdStatsNUcastOverflowOctets OBJECT-TYPE
       SYNTAX      Counter32
       UNITS "octets"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The number of times the associated
           smonVlanIdStatsNUcastOctets counter has overflowed."
       ::= { smonVlanIdStatsEntry 12 }

      smonVlanIdStatsNUcastHCOctets OBJECT-TYPE
       SYNTAX      Counter64
       UNITS "octets"
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of Non-unicast octets counted on
           this VLAN."
       ::= { smonVlanIdStatsEntry 13 }

      smonVlanIdStatsCreateTime OBJECT-TYPE
```

```
      SYNTAX      LastCreateTime
      MAX-ACCESS read-only
      STATUS      current
      DESCRIPTION
          "The value of sysUpTime when this entry was last
          activated. This object allows to a management station to
          detect deletion and recreation cycles between polls."
      ::= { smonVlanIdStatsEntry 14 }

   -- smonPrioStatsControlTable

   smonPrioStatsControlTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF SmonPrioStatsControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "Controls the setup of priority statistics tables.

         The smonPrioStatsControlTable allows configuration of
         collections based on the value of the 3-bit user priority
         field encoded in the Tag Control Information (TCI) field
         according to [19],[20].

         Note that this table merely reports priority as encoded in
         the VLAN headers, not the priority (if any) given to the
         frame for the actual switching purposes."

     ::= { smonStats 3 }

   smonPrioStatsControlEntry OBJECT-TYPE
    SYNTAX      SmonPrioStatsControlEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row in the smonPrioStatsControlTable."
    INDEX { smonPrioStatsControlIndex }
    ::= { smonPrioStatsControlTable 1 }


   SmonPrioStatsControlEntry ::= SEQUENCE {
    smonPrioStatsControlIndex                 Integer32,
    smonPrioStatsControlDataSource           DataSource,
    smonPrioStatsControlCreateTime        LastCreateTime,
    smonPrioStatsControlOwner                OwnerString,
    smonPrioStatsControlStatus                 RowStatus
   }

   smonPrioStatsControlIndex OBJECT-TYPE
```

```
   SYNTAX       Integer32 (1..65535)
   MAX-ACCESS   not-accessible
   STATUS       current
   DESCRIPTION
       "A unique arbitrary index for this smonPrioStatsControlEntry."
    ::= { smonPrioStatsControlEntry 1 }

  smonPrioStatsControlDataSource OBJECT-TYPE
   SYNTAX       DataSource
   MAX-ACCESS   read-create
   STATUS       current
   DESCRIPTION
       "The source of data for this set of VLAN statistics.

       This object MAY NOT be modified if the associated
       smonPrioStatsControlStatus object is equal to active(1)."
    ::= { smonPrioStatsControlEntry 2 }

  smonPrioStatsControlCreateTime OBJECT-TYPE
   SYNTAX      LastCreateTime
   MAX-ACCESS read-only
   STATUS       current
   DESCRIPTION
        "The value of sysUpTime when this entry was created.
       This object allows to a management station to
       detect deletion and recreation cycles between polls."

    ::= { smonPrioStatsControlEntry 3 }

  smonPrioStatsControlOwner OBJECT-TYPE
   SYNTAX       OwnerString
   MAX-ACCESS   read-create
   STATUS       current
   DESCRIPTION
        "Administratively assigned named of the owner of this entry.
       It usually defines the entity that created this entry and is
       therefore using the resources assigned to it, though there is
       no enforcement mechanism, nor assurance that rows created are
       ever used."
    ::= { smonPrioStatsControlEntry 4 }

  smonPrioStatsControlStatus OBJECT-TYPE
   SYNTAX       RowStatus
   MAX-ACCESS   read-create
   STATUS       current
   DESCRIPTION
       "The status of this row.
```

```
      An entry MAY NOT exist in the active state unless all
      objects in the entry have an appropriate value.

      If this object is not equal to active(1), all associated
      entries in the smonPrioStatsTable SHALL be deleted."
   ::= { smonPrioStatsControlEntry 5 }

   -- The Priority Statistics Table

   smonPrioStatsTable  OBJECT-TYPE
    SYNTAX      SEQUENCE OF SmonPrioStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "Contains the priority statistics. The collections are based
        on the value of the 3-bit user priority field encoded in the
        Tag Control Information (TCI) field according to [19], [20].
        Note that this table merely reports priority as encoded in
        the VLAN headers, not the priority (if any) given to the
        frame for the actual switching purposes.

        No garbage collection is designed for this table, as there
        always are at most eight rows per statistical set, and the
        low memory requirements do not justify the implementation of
        such a mechanism."
    ::= { smonStats 4 }

   smonPrioStatsEntry  OBJECT-TYPE
    SYNTAX      SmonPrioStatsEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
       "A conceptual row in smonPrioStatsTable."
    INDEX { smonPrioStatsControlIndex, smonPrioStatsId }
    ::= { smonPrioStatsTable 1 }

   SmonPrioStatsEntry ::= SEQUENCE {
    smonPrioStatsId                    Integer32,
    smonPrioStatsPkts                  Counter32,
    smonPrioStatsOverflowPkts          Counter32,
    smonPrioStatsHCPkts                Counter64,
    smonPrioStatsOctets                Counter32,
    smonPrioStatsOverflowOctets        Counter32,
    smonPrioStatsHCOctets              Counter64
   }

   smonPrioStatsId OBJECT-TYPE
    SYNTAX      Integer32 (0..7)
```

```
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The unique identifier of the priority level monitored for
         this specific statistics collection."
    REFERENCE
        " Draft Standard for Virtual Bridged Local Area Networks,
          P802.1Q/D10, chapter 4.3.2.1"
    ::= { smonPrioStatsEntry 1 }

  smonPrioStatsPkts OBJECT-TYPE
   SYNTAX      Counter32
   UNITS "packets"
   MAX-ACCESS read-only
   STATUS      current
   DESCRIPTION
       "The total number of packets counted on
        this priority level."
    ::= { smonPrioStatsEntry 2 }

  smonPrioStatsOverflowPkts OBJECT-TYPE
   SYNTAX      Counter32
   UNITS "packets"
   MAX-ACCESS read-only
   STATUS      current
   DESCRIPTION
       "The number of times the associated smonPrioStatsPkts
        counter has overflowed."
    ::= { smonPrioStatsEntry 3 }

  smonPrioStatsHCPkts OBJECT-TYPE
   SYNTAX      Counter64
   UNITS "packets"
   MAX-ACCESS read-only
   STATUS      current
   DESCRIPTION
       "The total number of packets counted on
        this priority level."
    ::= { smonPrioStatsEntry 4 }

  smonPrioStatsOctets OBJECT-TYPE
   SYNTAX      Counter32
   UNITS "octets"
   MAX-ACCESS read-only
   STATUS      current
   DESCRIPTION
       "The total number of octets counted on
        this priority level."
```

```
    ::= { smonPrioStatsEntry 5 }

smonPrioStatsOverflowOctets OBJECT-TYPE
 SYNTAX       Counter32
 UNITS "octets"
 MAX-ACCESS read-only
 STATUS       current
 DESCRIPTION
     "The number of times the associated smonPrioStatsOctets
      counter has overflowed."
 ::= { smonPrioStatsEntry 6 }

smonPrioStatsHCOctets OBJECT-TYPE
 SYNTAX       Counter64
 UNITS "octets"
 MAX-ACCESS read-only
 STATUS       current
 DESCRIPTION
     "The total number of octets counted on
      this priority level."
 ::= { smonPrioStatsEntry 7 }


portCopyTable  OBJECT-TYPE
 SYNTAX       SEQUENCE OF PortCopyEntry
 MAX-ACCESS  not-accessible
 STATUS       current
 DESCRIPTION
     " Port Copy provides the ability to copy all frames from a
      specified source to specified destination within a switch.
      Source and destinations MUST be ifEntries, as defined by [22].
      One to one, one to many, many to one and many to many source to
      destination relationships may be configured.

      Applicable counters on the destination will increment for all
      packets transiting the port, be it by normal bridging/switching
      or due to packet copy.
      Note that this table manages no RMON data collection by itself,
      and an agent may possibly implement no RMON objects except
      objects related to the port copy operation defined by the
      portCopyCompliance conformance macro. That allows for a switch
      with no other embedded RMON capability to perform port copy
      operations to a destination port at which a different external
      RMON probe is connected.

      One to one, many to one and one to many source to destination
      relationships may be configured.
```

        Each row that exists in this table defines such a
        relationship. By disabling a row in this table the port copy
        relationship no longer exists.

        The number of entries and the types of port copies (1-1,
        many-1, 1-many) are implementation specific and could
        possibly be dynamic due to changing resource availability.

        In order to configure a source to destination portCopy
        relationship, both source and destination interfaces MUST be
        present as an ifEntry in the ifTable and their respective
        ifAdminStatus and ifOperStatus values MUST be equal to
        'up(1)'. If the value of any of those two objects changes
        after the portCopyEntry is activated, portCopyStatus will
        transition to 'notReady(3)'.

        The capability of an interface to be source or destination of
        a port copy operation is described by the 'copySourcePort(0)'
        and 'copyDestPort(1)' bits in dataSourceCopyCaps. Those bits
        SHOULD be appropriately set by the agent, in order to allow
        for a portCopyEntry to be created.

        Applicable counters on the destination will increment for all
        packets transmitted, be it by normal bridging/switching or
        due to packet copy."
     ::= { portCopyConfig 1 }

   portCopyEntry   OBJECT-TYPE
    SYNTAX   PortCopyEntry
    MAX-ACCESS   not-accessible
    STATUS   current
    DESCRIPTION
        "Describes a particular port copy entry."
    INDEX { portCopySource, portCopyDest }
    ::= { portCopyTable 1 }

   PortCopyEntry ::= SEQUENCE {
        portCopySource
               InterfaceIndex,
        portCopyDest
               InterfaceIndex,
        portCopyDestDropEvents
               Counter32,
        portCopyDirection
               INTEGER,
        portCopyStatus
               RowStatus
        }

```
   portCopySource OBJECT-TYPE
    SYNTAX         InterfaceIndex
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
        "The ifIndex of the source which will have all packets
         redirected to the destination as defined by portCopyDest."
    ::= { portCopyEntry 1 }

   portCopyDest OBJECT-TYPE
    SYNTAX         InterfaceIndex
    MAX-ACCESS     not-accessible
    STATUS         current
    DESCRIPTION
        "Defines the ifIndex destination for the copy operation."
    ::= { portCopyEntry 2 }

   portCopyDestDropEvents OBJECT-TYPE
    SYNTAX         Counter32
    UNITS "events"
    MAX-ACCESS     read-only
    STATUS         current
    DESCRIPTION
         "The total number of events in which port copy packets were
         dropped by the switch at the destination port due to lack of
         resources.

         Note that this number is not necessarily the number of
         packets dropped; it is just the number of times this
         condition has been detected.

         A single dropped event counter is maintained for each
         portCopyDest. Thus all instances associated with a given
         portCopyDest will have the same portCopyDestDropEvents
         value."
      ::= { portCopyEntry 3 }

   portCopyDirection OBJECT-TYPE
    SYNTAX         INTEGER {
        copyRxOnly(1),
        copyTxOnly(2),
        copyBoth(3)
    }
    MAX-ACCESS     read-create
    STATUS         current
    DESCRIPTION
        "This object affects the way traffic is copied from a switch
           source port, for the indicated port copy operation.
```

          If this object has the value 'copyRxOnly(1)', then only
          traffic received on the indicated source port will be copied
          to the indicated destination port.

          If this object has the value 'copyTxOnly(2)', then only
          traffic transmitted out the indicated source port will be
          copied to the indicated destination port.

          If this object has the value 'copyBoth(3)', then all traffic
          received or transmitted on the indicated source port will be
          copied to the indicated destination port.

          The creation and deletion of instances of this object is
          controlled by the portCopyRowStatus object. Note that there
          is no guarantee that changes in the value of this object
          performed while the associated portCopyRowStatus object is
          equal to active will not cause traffic discontinuities in the
          packet stream."
      DEFVAL { copyBoth }
      ::= { portCopyEntry 4 }

      portCopyStatus OBJECT-TYPE
       SYNTAX   RowStatus
       MAX-ACCESS   read-create
       STATUS   current
       DESCRIPTION
          "Defines the status of the port copy entry.

          In order to configure a source to destination portCopy
          relationship, both source and destination interfaces MUST be
          present as an ifEntry in the ifTable and their respective
          ifAdminStatus and ifOperStatus values MUST be equal to
          'up(1)'. If the value of any of those two objects changes
          after the portCopyEntry is activated, portCopyStatus will
          transition to 'notReady(3)'.

          The capability of an interface to be source or destination of
          a port copy operation is described by the 'copySourcePort(0)'
          and 'copyDestPort(1)' bits in dataSourceCopyCaps. Those bits
          SHOULD be appropriately set by the agent, in order to allow
          for a portCopyEntry to be created."
       ::= { portCopyEntry 5 }

    -- smonRegistrationPoints
    -- defines a set of OIDs for registration purposes of entities
    -- supported by the SMON MIB.

    smonVlanDataSource

```
          OBJECT IDENTIFIER ::= { smonRegistrationPoints 1}

   -- Defined for use as an SmonDataSource. A single integer parameter
   -- is appended to the end of this OID when actually encountered in
   -- the dataSourceCapsTable, which represents a positive, non-zero
   -- VLAN identifier value.


   -- Conformance Macros

   smonMIBCompliances       OBJECT IDENTIFIER ::= { rmonConformance 3}
   smonMIBGroups            OBJECT IDENTIFIER ::= { rmonConformance 4}

   smonMIBCompliance        MODULE-COMPLIANCE
       STATUS        current
       DESCRIPTION
       "Describes the requirements for full conformance with the SMON
       MIB"
       MODULE -- this module
       MANDATORY-GROUPS                  {dataSourceCapsGroup,
                                          smonVlanStatsGroup,
                                          smonPrioStatsGroup,
                                          portCopyConfigGroup,
                                          smonInformationGroup}

       GROUP         smonHcTo100mbGroup
       DESCRIPTION
       "This group of VLAN statistics counter are mandatory only for
        those network interfaces for which the corresponding ifSpeed
        can be greater than 10MB/sec and less than or equal to
        100MB/sec."

       GROUP         smonHc100mbPlusGroup
       DESCRIPTION
       "This group of VLAN statistics counters are mandatory only for
        those network interfaces for which the corresponding ifSpeed
        can be more than 100MB/sec. This group of VLAN statistics is
        also mandatory for smonDataSources of type VLAN or
        entPhysicalEntry."

   ::= { smonMIBCompliances 1 }

   smonMIBVlanStatsCompliance         MODULE-COMPLIANCE
       STATUS        current
       DESCRIPTION
       "Describes the requirements for conformance with the SMON MIB
        with support for VLAN Statistics. Mandatory for a SMON probe
        in environment where IEEE 802.1Q bridging is implemented."
       MODULE -- this module
```

```
         MANDATORY-GROUPS                       {dataSourceCapsGroup,
                                                 smonVlanStatsGroup,
                                                 smonInformationGroup}
         GROUP           hcVlanTo100mbGroup
         DESCRIPTION
         "This group of VLAN statistics counter are mandatory only
          for those network interfaces for which the corresponding
          ifSpeed can be up to and including 100MB/sec."

         GROUP           hcVlan100mbPlusGroup
         DESCRIPTION
         "This group of VLAN statistics counters are mandatory only for
          those network interfaces for which the corresponding ifSpeed
          is greater than 100MB/sec. This group of VLAN statistics is
          also mandatory for smonDataSources of type VLAN or
          entPhysicalEntry."

     ::= { smonMIBCompliances 2 }

    smonMIBPrioStatsCompliance          MODULE-COMPLIANCE
         STATUS          current
         DESCRIPTION
         "Describes the requirements for conformance with the SMON MIB
          with support for priority level Statistics. Mandatory for a
          SMON probe in a environment where IEEE 802.1p
          priority-switching is implemented."
         MODULE -- this module
         MANDATORY-GROUPS                       {dataSourceCapsGroup,
                                                 smonPrioStatsGroup,
                                                 smonInformationGroup}

         GROUP           hcPrioTo100mbGroup
         DESCRIPTION
         "This group of VLAN priority statistics counters are mandatory
          only for those network interfaces for which the corresponding
          ifSpeed can be up to and including 100MB/sec."

         GROUP           hcPrio100mbPlusGroup
         DESCRIPTION
         "This group is mandatory only for those network
          interfaces for which the corresponding ifSpeed is greater
          than 100MB/sec. This group of VLAN priority
          statistics is also mandatory for smonDataSources of type
          VLAN or entPhysicalEntry"

     ::= { smonMIBCompliances 3 }

    portCopyCompliance                  MODULE-COMPLIANCE
```

```
        STATUS        current
        DESCRIPTION
        "Describes the requirements for conformance with the port copy
         functionality defined by the SMON MIB"
        MODULE  -- this module
        MANDATORY-GROUPS                    {dataSourceCapsGroup,
                                             portCopyConfigGroup,
                                             smonInformationGroup}

         ::= { smonMIBCompliances 4}

    dataSourceCapsGroup      OBJECT-GROUP
        OBJECTS            { dataSourceRmonCaps,
                             dataSourceCopyCaps,
                             dataSourceCapsIfIndex}
        STATUS        current
        DESCRIPTION
        "Defines the objects that describe the capabilities of RMON
         data sources."
     ::= {smonMIBGroups 1 }

    smonVlanStatsGroup       OBJECT-GROUP
        OBJECTS            { smonVlanStatsControlDataSource,
                             smonVlanStatsControlCreateTime,
                             smonVlanStatsControlOwner,
                             smonVlanStatsControlStatus,
                             smonVlanIdStatsTotalPkts,
                             smonVlanIdStatsTotalOctets,
                             smonVlanIdStatsNUcastPkts,
                             smonVlanIdStatsCreateTime}
        STATUS        current
        DESCRIPTION
        "Defines the switch monitoring specific statistics - per VLAN
         Id on interfaces of 10MB or less."
     ::= { smonMIBGroups 2 }

    smonPrioStatsGroup       OBJECT-GROUP
        OBJECTS            { smonPrioStatsControlDataSource,
                             smonPrioStatsControlCreateTime,
                             smonPrioStatsControlOwner,
                             smonPrioStatsControlStatus,
                             smonPrioStatsPkts,
                             smonPrioStatsOctets}
        STATUS        current
        DESCRIPTION
        "Defines the switch monitoring specific statistics - per VLAN
         Id on interface."
```

```
    ::= { smonMIBGroups 3 }

    smonHcTo100mbGroup       OBJECT-GROUP
        OBJECTS           { smonVlanIdStatsTotalOverflowOctets,
                            smonVlanIdStatsTotalHCOctets,
                            smonPrioStatsOverflowOctets,
                            smonPrioStatsHCOctets}
        STATUS      current
        DESCRIPTION
        "Defines the additional high capacity statistics needed to be
         kept on interfaces with ifSpeed greater than 10MB/sec and
         less than or equal to 100MB/sec."
    ::= { smonMIBGroups 4 }

    smonHc100mbPlusGroup     OBJECT-GROUP
        OBJECTS           { smonVlanIdStatsTotalOverflowPkts,
                            smonVlanIdStatsTotalHCPkts,
                            smonVlanIdStatsTotalOverflowOctets,
                            smonVlanIdStatsTotalHCOctets,
                            smonVlanIdStatsNUcastOverflowPkts,
                            smonVlanIdStatsNUcastHCPkts,
                            smonPrioStatsOverflowPkts,
                            smonPrioStatsHCPkts,
                            smonPrioStatsOverflowOctets,
                            smonPrioStatsHCOctets}
        STATUS      current
        DESCRIPTION
        "Defines the additional high capacity statistics needed to be
         kept on interfaces with ifSpeed of more than 100MB/sec. These
         statistics MUST also be kept on smonDataSources of type VLAN
         or entPhysicalEntry."
    ::= { smonMIBGroups 5 }

    hcVlanTo100mbGroup       OBJECT-GROUP
        OBJECTS           { smonVlanIdStatsTotalOverflowOctets,
                            smonVlanIdStatsTotalHCOctets}
        STATUS      current
        DESCRIPTION
        "Defines the additional high capacity VLAN statistics
         needed to be kept on interfaces with ifSpeed greater than
         10MB/sec and less than or equal to 100MB/sec."
    ::= { smonMIBGroups 6 }

    hcVlan100mbPlusGroup      OBJECT-GROUP
        OBJECTS           { smonVlanIdStatsTotalOverflowPkts,
                            smonVlanIdStatsTotalHCPkts,
                            smonVlanIdStatsTotalOverflowOctets,
                            smonVlanIdStatsTotalHCOctets,
```

```
                        smonVlanIdStatsNUcastOverflowPkts,
                        smonVlanIdStatsNUcastHCPkts}
        STATUS         current
        DESCRIPTION
        "Defines the additional high capacity VLAN statistics
         needed to be kept on interfaces with ifSpeed of more than
         100MB/sec.  These statistics MUST also be kept on
         smonDataSources of type VLAN or entPhysicalEntry."
    ::= { smonMIBGroups 7 }

    hcPrioTo100mbGroup        OBJECT-GROUP
        OBJECTS          { smonPrioStatsOverflowOctets,
                           smonPrioStatsHCOctets }
        STATUS         current
        DESCRIPTION
        "Defines the additional high capacity VLAN priority
         statistics needed to be kept on interfaces with
         ifSpeed of greater than 10MB/sec and less than or equal
         to 100MB/sec."
    ::= { smonMIBGroups 8 }

    hcPrio100mbPlusGroup        OBJECT-GROUP
        OBJECTS          { smonPrioStatsOverflowPkts,
                           smonPrioStatsHCPkts,
                           smonPrioStatsOverflowOctets,
                           smonPrioStatsHCOctets}
        STATUS         current
        DESCRIPTION
        "Defines the additional high capacity VLAN priority
         statistics needed to be kept on interfaces with
         ifSpeed of greater than 100MB/sec. These statistics MUST
         also be kept on smonDataSources of type VLAN or
         entPhysicalEntry."
    ::= { smonMIBGroups 9 }

    smonVlanStatsExtGroup   OBJECT-GROUP
        OBJECTS          {smonVlanIdStatsNUcastOctets,
                          smonVlanIdStatsNUcastOverflowOctets,
                          smonVlanIdStatsNUcastHCOctets}
        STATUS         current
        DESCRIPTION
        "Defines the switch monitoring specific statistics for systems
         capable of counting non-unicast octets for a given dataSource
         (as described in the dataSourceRmonCaps object)."
    ::= { smonMIBGroups 10 }

    smonInformationGroup    OBJECT-GROUP
        OBJECTS          { smonCapabilities }
```

```
        STATUS          current
        DESCRIPTION
        "An indication of the SMON capabilities supported by this
        agent."
    ::= { smonMIBGroups 11 }

    portCopyConfigGroup      OBJECT-GROUP
        OBJECTS            { portCopyDestDropEvents,
                             portCopyDirection,
                             portCopyStatus
                           }
        STATUS          current
        DESCRIPTION
        "Defines the control objects for copy port operations."
    ::= { smonMIBGroups 12 }

    END
```

6. References

    [1] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for
         Describing SNMP Management Frameworks", RFC 2571, April 1999.

    [2] Rose, M. and K. McCloghrie, "Structure and Identification of
         Management Information for TCP/IP-based Internets", STD 16, RFC
         1155, May 1990.

    [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16,
         RFC 1212, March 1991.

    [4] Rose, M., "A Convention for Defining Traps for use with the
         SNMP", RFC 1215, March 1991.

    [5] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose,
         M. and S. Waldbusser, "Structure of Management Information
         Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

    [6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose,
         M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58,
         RFC 2579, April 1999.

    [7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose,
         M. and S. Waldbusser, "Conformance Statements for SMIv2", STD
         58, RFC 2580, April 1999.

    [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple
         Network Management Protocol", STD 15, RFC 1157, May 1990.

   [9] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S.
        Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901,
        January 1996.

   [10] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S.
        Waldbusser, "Transport Mappings for Version 2 of the Simple
        Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

   [11] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message
        Processing and Dispatching for the Simple Network Management
        Protocol (SNMP)", RFC 2572, April 1999.

   [12] Blumenthal, U., and B. Wijnen, "User-based Security Model for
        Version 3 of the Simple Network Management Protocol (SNMPv3)",
        RFC 2574, April 1999.

   [13] SNMPv2 Working Group, Case, J., McCloghrie, K., Rose, M. and S.
        Waldbusser, "Protocol Operations for Version 2 of the Simple
        Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

   [14] Levi, D., Meyer, P., and B. Stewart, "SNMP Applications", RFC
        2573, April 1999.

   [15] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access
        Control Model for the Simple Network Management Protocol
        (SNMP)", RFC 2575, April 1999.

   [16] Waldbusser, S., "Remote Network Monitoring Management
        Information Base Version 2 using SMIv2", RFC 2021, January 1997.

   [17] Waldbusser, S., "Remote Network Monitoring Management
        Information Base", RFC 1757, February 1995.

   [18] McCloghrie, K. and A. Bierman, "Entity MIB", RFC 2037, October
        1996.

   [19] ISO/IEC Final CD 15802-3, ANSI/IEEE Std 802.1D-1998 "Information
        technology - Telecommunications and information exchange between
        systems - Local and metropolitan area networks - Common
        specifications – Part 3: Media Access Control (MAC) Bridges:
        Revision (Incorporating IEEE P802.1p: Traffic Class Expediting
        and Dynamic Multicast Filtering)", March 1998.

   [20] ANSI/IEEE Draft Standard P802.1Q/D10, "IEEE Standards for Local
        and Metropolitan Area Networks: Virtual Bridged Local Area
        Networks", March 1998.

   [21] De Graaf, K., Romascanu, D., McMaster, D. and K. McCloghrie,
        "Definition of Managed Objects for IEEE 802.3 Repeater Devices
        using SMIv2", RFC 2108, February 1997.

   [22] McCloghrie, K. and F. Kastenholz," The Interfaces Group MIB
        using SMIv2", RFC 2233, November 1997.

   [23] Decker, E. Langille, P., Rijsinghani, A. and K. McCloghrie.. -
        "Definitions of Managed Objects for Bridges", RFC 1493, July
        1993

   [24] Bradner, S., "Key words for use in RFCs to Indicate Requirement
        Levels", BCP 14, RFC 2119, March 1997.

   [25] McCloghrie, K. and M. Rose, Editors, "Management Information
        Base for Network Management of TCP/IP-based internets: MIB-II",
        STD 17, RFC 1213, March 1991.

7. Intellectual Property

   The IETF takes no position regarding the validity or scope of any
   intellectual property or other rights that might be claimed to
   pertain to the implementation or use of the technology described in
   this document or the extent to which any license under such rights
   might or might not be available; neither does it represent that it
   has made any effort to identify any such rights. Information on the
   IETF's procedures with respect to rights in standards-track and
   standards-related documentation can be found in BCP-11. Copies of
   claims of rights made available for publication and any assurances of
   licenses to be made available, or the result of an attempt made to
   obtain a general license or permission for the use of such
   proprietary rights by implementors or users of this specification can
   be obtained from the IETF Secretariat.

   The IETF invites any interested party to bring to its attention any
   copyrights, patents or patent applications, or other proprietary
   rights which may cover technology that may be required to practice
   this standard. Please address the information to the IETF Executive
   Director.

8. Security Considerations

   There are a number of management objects defined in this MIB that
   have 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.

There are a number of managed objects in this MIB that may contain
sensitive information. These are:

    smonCapabilities
    dataSourceCapsTable
    portCopyTable

It is thus important to control even GET access to these objects and
possibly to even encrypt the values of these object when sending them
over the network via SNMP. Not all versions of SNMP provide features
for such a secure environment.

SNMPv1 by itself is not a secure environment. 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.

It is RECOMMENDED that the implementors consider the security
features as provided by the SNMPv3 framework. Specifically, the use
of the User-based Security Model RFC 2574 [12] and the View-based
Access Control Model RFC 2575 [15] is RECOMMENDED.

It is then a customer/user responsibility to ensure that the SNMP
entity giving access to an instance of this MIB, is properly
configured to give access to the objects only to those principals
(users) that have legitimate rights to indeed GET or SET
(change/create/delete) them.

9. Authors' Addresses

Richard Waterman
Allot Communications
292 E. Main St.
Los Gatos, CA. 95030
USA

Phone: +1-408-399-3154
EMail: rich@allot.com

Bill Lahaye
Xylan Corporation
26707 W. Agoura Rd.
Calabasas, CA 91302
USA

Phone: +1-800-995-2612
EMail bill.lahaye@xylan.com

Dan Romascanu
Lucent Technologies
Atidim Technology Park, Bldg. #3
Tel Aviv, 61131
Israel

Phone: +972-3-645-8414
EMail: dromasca@lucent.com

Steven Waldbusser
International Network Services (INS)
1213 Innsbruck Dr.
Sunnyvale, CA 94089

Phone: +1-650-318-1251
EMail: waldbusser@ins.com

A. Full Copyright Statement

Acknowledgement