Network Working Group
Request for Comments: 3144
Category: Standards Track

D. Romascanu
Avaya, Inc.

August 2001

Remote Monitoring MIB Extensions for Interface Parameters Monitoring

Status of this Memo

Copyright Notice

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. The document proposes an extension to the Remote Monitoring MIB with a method of sorting the interfaces of a monitored device according to values of parameters specific to this interface.

Table of Contents

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it defines a method of sorting the interfaces of a monitored device according to values of parameters specific to this interface.

This memo also includes a MIB module. This MIB module extends the list of managed objects specified in [RFC2819] and [RFC2613].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMEND", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [RFC2571].
- 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 [RFC1155], STD 16, RFC 1212 [RFC1212] and RFC 1215 [RFC1215]. The second version, called SMIv2, is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [RFC1157]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [RFC1901] and RFC 1906 [RFC1906]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [RFC1906], RFC 2572 [RFC2572] and RFC 2574 [RFC2574].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [RFC1157]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [RFC1905].
- A set of fundamental applications described in RFC 2573 [RFC2573] and the view-based access control mechanism described in RFC 2575 [RFC2575].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [RFC2570].

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 translation is possible (use of Counter64). Some machine readable 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.

3. Overview

This document continues the architecture created in the RMON MIB [RFC2819] and extended by the SMON MIB [RFC2613] by providing a method of ordering the interfaces of a device according to the value of a specific parameter that characterizes the interfaces.

The need for such a technique derives from the evolution of the network devices - bridges, routers, etc., into complex entities with a large number of interfaces and with many parameters that need to be monitored on each interface. It is common for certain classes of switching devices to contain hundred of ports, and for each port to instrument and support tens of parameters - usually expressed as counters - for each interface. As a result, it becomes impossible for applications that monitor these devices to provide a view that would allow the user to understand easily what is the status of the device, whether the behavior of a port or interface is in normal boundaries or not, and which are the most congested or problematic interfaces of the device.

This document tries to answer this problem by proposing a method of providing a sorted list of interfaces according to programmable criteria. The result of applying this method will be a shorter list, that includes the most significant interfaces sorted according to the selected criteria. One possible action that can be taken by a network manager could be applying to this interface a copy port operation to a destination port that has a dedicated monitoring device (e.g., a network analyzer) connected to it. A standard MIB interface for performing this operation is described in [RFC2613].

4. MIB Structure

This MIB contains one MIB group:

- The interfaceTopNObjects

The interfaceTopNObjects includes one capability object and two tables:

- The interfaceControlTable
- The interfaceTopNTable

The interfaceControlTable is an RMON-style control table, allowing for the creation of interfaceTopN reports. The parameters specific for each report, like the duration of the report, the number of reports, start time and the characteristics of the variables that are sorted (absolute, 'deltas' or percentage of the total bandwidth) are set in this table. An optional operation that is controlled from this table is the normalization of values of the variables, which allows for sorting of variables on the interfaces, despite the basic speed of the interfaces being different on different interfaces.

5. Evolution of the Document, Limitations and Future Work

The RMON MIB Working Group included in its Charter a MIB document that would offer a solution to the problem of quickly determining the most congested (highest utilized) physical ports and links in an RMON-capable device with multiple interfaces.

An initial solution, proposed in the first version of this document included a limited approach. The objects whose values are used as criteria for sorting are elements in tables indexed by an InterfaceIndex type of object, as defined in [RFC2863]. This approach simplifies the search algorithm and the result table, but restricts the method to interface parameters. A more generic ' usrTopN' function was initially considered out of the scope of this work.

At the Working Group meeting in Adelaide in March 2000, it was decided to try to define the more generic function of usrTopN. Under this approach, variables belonging to tables with any type of index can be sorted, but at expense of extra processing and sanity checking by the agent.

At the interim meeting of the RMON Working Group in San Francisco, in May 2000, it was decided that the usrTopN solution would not be continued in this phase of the work. One of the reasons is that it is difficult to achieve a normalization factor for a generic

approach. The group agreed it is not desirable to require the application to plug-in the scaling factor for every instance that might be included in a TopN report.

6. Definitions

```
INTERFACETOPN-MIB DEFINITIONS ::= BEGIN

   IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE, Integer32, Gauge32
                FROM SNMPv2-SMI
        RowStatus, TimeStamp, TruthValue
                FROM SNMPv2-TC
        rmon, OwnerString
                FROM RMON-MIB
        CounterBasedGauge64
                FROM HCNUM-TC
        MODULE-COMPLIANCE, OBJECT-GROUP
                FROM SNMPv2-CONF;

   interfaceTopNMIB MODULE-IDENTITY
        LAST-UPDATED "200103270000Z"
        ORGANIZATION "IETF RMON MIB Working Group"
        CONTACT-INFO
        "

                Dan Romascanu
                Avaya Inc.
                Tel:  +972-3-645-8414
                Email: dromasca@avaya.com"

       DESCRIPTION
          "The MIB module for sorting device interfaces for RMON and
          SMON monitoring in a multiple device implementation."
      ::= { rmon 27 }
interfaceTopNObjects       OBJECT IDENTIFIER ::= { interfaceTopNMIB 1 }
interfaceTopNNotifications OBJECT IDENTIFIER ::= { interfaceTopNMIB 2 }
interfaceTopNConformance   OBJECT IDENTIFIER ::= { interfaceTopNMIB 3 }

-- The Interface Top N group is used to prepare reports that
-- describe a list of interfaces (data sources)
-- ordered by the values of one
-- of the objects that apply to the interfaces of the respective device.
-- Those objects are defined by standard MIBs. The exact objects that
-- are supported by the agent are described by interfaceTopNCaps
-- The objects must be elements in tables indexed only by an
-- InterfaceIndex object.
-- The objects  chosen by the
```

```
-- management station may be sampled over a management
-- station-specified time interval, making the report rate based.
-- The management station also specifies the number of interfaces
-- that are reported.
--
-- The interfaceTopNControlTable is used to initiate the generation
-- of a report.  The management station may select the parameters
-- of such a report, such as which object, how
-- many interfaces, and the start & stop times of the sampling.  When
-- the report is prepared, entries are created in the
-- interfaceTopNTable associated with the relevant
-- interfaceTopNControlEntry.  These entries are static for
-- each report after it has been prepared.

interfaceTopNCaps   OBJECT-TYPE
    SYNTAX      BITS {
                ifInOctets(0),
                ifInUcastPkts(1),
                ifInNUcastPkts(2),
                ifInDiscards(3),
                ifInErrors(4),
                ifInUnknownProtos(5),
                ifOutOctets(6),
                ifOutUcastPkts(7),
                ifOutNUcastPkts(8),
                ifOutDiscards(9),
                ifOutErrors(10),
                ifInMulticastPkts(11),
                ifInBroadcastPkts(12),
                ifOutMulticastPkts(13),
                ifOutBroadcastPkts(14),
                ifHCInOctets(15),
                ifHCInUcastPkts(16),
                ifHCInMulticastPkts(17),
                ifHCInBroadcastPkts(18),
                ifHCOutOctets(19),
                ifHCOutUcastPkts(20),
                ifHCOutMulticastPkts(21),
                ifHCOutBroadcastPkts(22),
                dot3StatsAlignmentErrors(23),
                dot3StatsFCSErrors(24),
                dot3StatsSingleCollisionFrames(25),
                dot3StatsMultipleCollisionFrames(26),
                dot3StatsSQETestErrors(27),
                dot3StatsDeferredTransmissions(28),
                dot3StatsLateCollisions(29),
                dot3StatsExcessiveCollisions(30),
                dot3StatsInternalMacTxErrors(31),
```

```
                dot3StatsCarrierSenseErrors(32),
                dot3StatsFrameTooLongs(33),
                dot3StatsInternalMacRxErrors(34),
                dot3StatsSymbolErrors(35),
                dot3InPauseFrames(36),
                dot3OutPauseFrames(37),
                dot5StatsLineErrors(38),
                dot5StatsBurstErrors(39),
                dot5StatsACErrors(40),
                dot5StatsAbortTransErrors(41),
                dot5StatsInternalErrors(42),
                dot5StatsLostFrameErrors(43),
                dot5StatsReceiveCongestions(44),
                dot5StatsFrameCopiedErrors(45),
                dot5StatsTokenErrors(46),
                dot5StatsSoftErrors(47),
                dot5StatsHardErrors(48),
                dot5StatsSignalLoss(49),
                dot5StatsTransmitBeacons(50),
                dot5StatsRecoverys(51),
                dot5StatsLobeWires(52),
                dot5StatsRremoves(53),
                dot5StatsSingles(54),
                dot5StatsFreqErrors(55),
                etherStatsDropEvents(56),
                etherStatsOctets(57),
                etherStatsPkts(58),
                etherStatsBroadcastPkts(59),
                etherStatsMulticastPkts(60),
                etherStatsCRCAlignErrors(61),
                etherStatsUndersizePkts(62),
                etherStatsOversizePkts(63),
                etherStatsFragments(64),
                etherStatsJabbers(65),
                etherStatsCollisions(66),
                etherStatsPkts64Octets(67),
                etherStatsPkts65to127Octets(68),
                etherStatsPkts128to255Octets(69),
                etherStatsPkts256to511Octets(70),
                etherStatsPkts512to1023Octets(71),
                etherStatsPkts1024to1518Octets(72),
                dot1dTpPortInFrames(73),
                dot1dTpPortOutFrames(74),
                dot1dTpPortInDiscards(75)
    }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

"The type(s) of sorting capabilities supported by the agent.

If the agent can perform sorting of interfaces according to the
values of ifInOctets, as defined in [RFC2863],
then the 'ifInOctets' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifInUcastPkts, as defined in [RFC2863],
then the 'ifInUcastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifInNUcastPkts, as defined in [RFC2863],
then the 'ifInNUcastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifInDiscards, as defined in [RFC2863],
then the 'ifInDiscards' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifInErrors, as defined in [RFC2863],
then the 'ifInErrors' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifInUnknownProtocols, as defined in [RFC2863],
then the 'ifInUnknownProtocols' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifOutOctets, as defined in [RFC2863],
then the 'ifOutOctets' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifOutUcastPackets, as defined in [RFC2863],
then the 'ifOutUcastPackets' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifOutNUcastPackets, as defined in [RFC2863],
then the 'ifOutNUcastPackets' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifOutDiscards, as defined in [RFC2863],
then the 'ifOutDiscards' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifOutErrors, as defined in [RFC2863],
then the 'ifOutErrors' bit will be set.

If the agent can perform sorting of interfaces according to the
values of ifInMulticastPkts, as defined in [RFC2863],

then the 'ifInMulticastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifInBroadcastPkts, as defined in [RFC2863], then the 'ifInBroadcastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifOutMulticastPkts, as defined in [RFC2863], then the 'ifOutMulticastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifOutBroadcastPkts, as defined in [RFC2863], then the 'ifOutBroadcastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifHCInOctets, as defined in [RFC2863], then the 'ifHCInOctets' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifHCInMulticastPkts, as defined in [RFC2863], then the 'ifHCInMulticastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifHCInBroadcastPkts, as defined in [RFC2863], then the 'ifHCInBroadcastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifHCOutOctets, as defined in [RFC2863], then the 'ifHCOutOctets' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifHCOutUcastPkts, as defined in [RFC2863], then the 'ifHCOutUcastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifHCOutMulticastPkts, as defined in [RFC2863], then the 'ifHCOutMulticastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of ifHCOutBroadcastPkts, as defined in [RFC2863], then the 'ifHCOutBroadcastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsAlignmentErrors, as defined in [RFC2665], then the 'dot3StatsAlignmentErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsFCSErrors, as defined in [RFC2665],

then the 'dot3StatsFCSErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsSingleCollisionFrames, as defined in [RFC2665],then the 'dot3StatsSingleCollisionFrames' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsSQETestErrors, as defined in [RFC2665], then the 'dot3StatsSQETestErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsDeferredTransmissions, as defined in [RFC2665], then the 'dot3StatsDeferredTransmissions' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsLateCollisions, as defined in [RFC2665], then the 'dot3StatsLateCollisions' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsExcessiveCollisions, as defined in [RFC2665], then the 'dot3StatsExcessiveCollisions' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsInternalMacTxErrors, as defined in [RFC2665],then the 'dot3StatsInternalMacTxErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsCarrierSenseErrors, as defined in [RFC2665], then the 'dot3StatsCarrierSenseErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsFrameTooLongs, as defined in [RFC2665], then the 'dot3StatsFrameTooLongs' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsInternalMacRxErrors, as defined in [RFC2665], then the 'dot3StatsInternalMacRxErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3StatsSymbolErrors, as defined in [RFC2665], then the 'dot3StatsSymbolErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3InPauseFrames, as defined in [RFC2665],

then the 'dot3InPauseFrames' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot3OutPauseFrames, as defined in [RFC2665], then the 'dot3OutPauseFrames' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsLineErrors, as defined in [RFC1748], then the 'dot5StatsLineErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsBurstErrors, as defined in [RFC1748], then the 'dot5StatsBurstErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsACErrors, as defined in [RFC1748], then the 'dot5StatsACErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsAbortTransErrors, as defined in [RFC1748], then the 'dot5StatsAbortTransErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsInternalErrors, as defined in [RFC1748], then the 'dot5StatsInternalErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsLostFrameErrors, as defined in [RFC1748], then the 'dot5StatsLostFrameErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsReceiveCongestionErrors, as defined in [RFC1748], then the 'dot5StatsReceiveCongestionErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsFrameCopiedErrors, as defined in [RFC1748], then the 'dot5StatsFrameCopiedErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsTokenErrors, as defined in [RFC1748], then the 'dot5StatsTokenErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of dot5StatsSoftErrors, as defined in [RFC1748], then the 'dot5StatsSoftErrors' bit will be set.

If the agent can perform sorting of interfaces according to the

values of dot5StatsHardErrors, as defined in [RFC1748],
then the 'dot5StatsHardErrors' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot5StatsSignalLoss, as defined in [RFC1748],
then the 'dot5StatsSignalLoss' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot5StatsTransmitBeacons, as defined in [RFC1748],
then the 'dot5StatsTransmitBeacons' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot5StatsRecoverys, as defined in [RFC1748],
then the 'dot5StatsRecoverys' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot5StatsLobeWires, as defined in [RFC1748],
then the 'dot5StatsLobeWires' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot5StatsRemoves, as defined in [RFC1748],
then the 'dot5StatsRemoves' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot5StatsSingles, as defined in [RFC1748],
then the 'dot5StatsSingles' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot5StatsFreqErrors, as defined in [RFC1748],
then the 'dot5StatsFreqErrors' bit will be set.

If the agent can perform sorting of interfaces according to the
values of etherStatsDropEvents, as defined in [RFC2819],
then the 'etherStatsDropEvents' bit will be set.

If the agent can perform sorting of interfaces according to the
values of etherStatsOctets, as defined in [RFC2819],
then the 'etherStatsOctets' bit will be set.

If the agent can perform sorting of interfaces according to the
values of etherStatsPkts, as defined in [RFC2819],
then the 'etherStatsPkts' bit will be set.

If the agent can perform sorting of interfaces according to the
values of etherStatsBroadcastPkts, as defined in [RFC2819],
then the 'etherStatsBroadcastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the

values of etherStatsMulticastPkts, as defined in [RFC2819], then the 'etherStatsMulticastPkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsCRCAlignErrors, as defined in [RFC2819], then the 'etherStatsCRCAlignErrors' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsUndersizePkts, as defined in [RFC2819], then the 'etherStatsUndersizePkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsOversizePkts, as defined in [RFC2819], then the 'etherStatsOversizePkts' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsFragments, as defined in [RFC2819], then the 'etherStatsFragments' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsJabbers, as defined in [RFC2819], then the 'etherStatsJabbers' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsCollisions, as defined in [RFC2819], then the 'etherStatsCollisions' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsPkts64Octets, as defined in [RFC2819], then the 'etherStatsPkts64Octets' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsPkts65to127Octets, as defined in [RFC2819], then the 'etherStatsPkts65to127Octets' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsPkts128to255Octets, as defined in [RFC2819], then the 'etherStatsPkts128to255Octets' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsPkts256to511Octets, as defined in [RFC2819], then the 'etherStatsPkts256to511Octets' bit will be set.

If the agent can perform sorting of interfaces according to the values of etherStatsPkts512to1023Octets, as defined in [RFC2819], then the 'etherStatsPkts512to1023Octets' bit will be set.

If the agent can perform sorting of interfaces according to the

```
values of etherStatsPkts1024to1518Octets, as defined in
[RFC2819], then the 'etherStatsPkts1024to1518Octets' bit will
be set.


If the agent can perform sorting of interfaces according to the
values of dot1dTpPortInFrames, as defined in [RFC1493],
then the 'dot1dTpPortInFrames' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot1dTpPortOutFrames, as defined in [RFC1493],
then the 'dot1dTpPortOutFrames' bit will be set.

If the agent can perform sorting of interfaces according to the
values of dot1dTpPortInDiscards, as defined in [RFC1493],
then the 'dot1dTpPortInDiscards' bit will be set."

    ::= { interfaceTopNObjects 1 }



  interfaceTopNControlTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF InterfaceTopNControlEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
        "A table of control records for reports on the top 'N'
        interfaces for the value or rate of a selected object.
        The number of entries depends on the configuration of the agent.
        The maximum number of entries is implementation
        dependent."
     ::= { interfaceTopNObjects 2 }

   interfaceTopNControlEntry OBJECT-TYPE
       SYNTAX      InterfaceTopNControlEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
               "A set of parameters that control the creation of a
               report of the top N ports according to several metrics."
       INDEX    { interfaceTopNControlIndex }
       ::= { interfaceTopNControlTable 1 }

   InterfaceTopNControlEntry ::= SEQUENCE {
       interfaceTopNControlIndex
           Integer32,
       interfaceTopNObjectVariable
           INTEGER,
```

```
        interfaceTopNObjectSampleType
            INTEGER,
        interfaceTopNNormalizationReq
            TruthValue,
        interfaceTopNNormalizationFactor
            Integer32,
        interfaceTopNTimeRemaining
            Integer32,
        interfaceTopNDuration
            Integer32,
        interfaceTopNRequestedSize
            Integer32,
        interfaceTopNGrantedSize
            Integer32,
        interfaceTopNStartTime
            TimeStamp,
        interfaceTopNOwner
            OwnerString,
        interfaceTopNLastCompletionTime
            TimeStamp,
        interfaceTopNRowStatus
            RowStatus
    }

    interfaceTopNControlIndex OBJECT-TYPE
        SYNTAX      Integer32 (1 .. 65535)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "An index that uniquely identifies an entry in the
           interfaceTopNControl table.  Each such entry defines
           one top N report prepared for a probe."
        ::= { interfaceTopNControlEntry 1 }


interfaceTopNObjectVariable OBJECT-TYPE
    SYNTAX INTEGER   {
                ifInOctets(0),
                ifInUcastPkts(1),
                ifInNUcastPkts(2),
                ifInDiscards(3),
                ifInErrors(4),
                ifInUnknownProtos(5),
                ifOutOctets(6),
                ifOutUcastPkts(7),
                ifOutNUcastPkts(8),
                ifOutDiscards(9),
                ifOutErrors(10),
```

```
                ifInMulticastPkts(11),
                ifInBroadcastPkts(12),
                ifOutMulticastPkts(13),
                ifOutBroadcastPkts(14),
                ifHCInOctets(15),
                ifHCInUcastPkts(16),
                ifHCInMulticastPkts(17),
                ifHCInBroadcastPkts(18),
                ifHCOutOctets(19),
                ifHCOutUcastPkts(20),
                ifHCOutMulticastPkts(21),
                ifHCOutBroadcastPkts(22),
                dot3StatsAlignmentErrors(23),
                dot3StatsFCSErrors(24),
                dot3StatsSingleCollisionFrames(25),
                dot3StatsMultipleCollisionFrames(26),
                dot3StatsSQETestErrors(27),
                dot3StatsDeferredTransmissions(28),
                dot3StatsLateCollisions(29),
                dot3StatsExcessiveCollisions(30),
                dot3StatsInternalMacTxErrors(31),
                dot3StatsCarrierSenseErrors(32),
                dot3StatsFrameTooLongs(33),
                dot3StatsInternalMacRxErrors(34),
                dot3StatsSymbolErrors(35),
                dot3InPauseFrames(36),
                dot3OutPauseFrames(37),
                dot5StatsLineErrors(38),
                dot5StatsBurstErrors(39),
                dot5StatsACErrors(40),
                dot5StatsAbortTransErrors(41),
                dot5StatsInternalErrors(42),
                dot5StatsLostFrameErrors(43),
                dot5StatsReceiveCongestions(44),
                dot5StatsFrameCopiedErrors(45),
                dot5StatsTokenErrors(46),
                dot5StatsSoftErrors(47),
                dot5StatsHardErrors(48),
                dot5StatsSignalLoss(49),
                dot5StatsTransmitBeacons(50),
                dot5StatsRecoverys(51),
                dot5StatsLobeWires(52),
                dot5StatsRemoves(53),
                dot5StatsSingles(54),
                dot5StatsFreqErrors(55),
                etherStatsDropEvents(56),
                etherStatsOctets(57),
                etherStatsPkts(58),
```

```
                etherStatsBroadcastPkts(59),
                etherStatsMulticastPkts(60),
                etherStatsCRCAlignErrors(61),
                etherStatsUndersizePkts(62),
                etherStatsOversizePkts(63),
                etherStatsFragments(64),
                etherStatsJabbers(65),
                etherStatsCollisions(66),
                etherStatsPkts64Octets(67),
                etherStatsPkts65to127Octets(68),
                etherStatsPkts128to255Octets(69),
                etherStatsPkts256to511Octets(70),
                etherStatsPkts512to1023Octets(71),
                etherStatsPkts1024to1518Octets(72),
                dot1dTpPortInFrames(73),
                dot1dTpPortOutFrames(74),
                dot1dTpPortInDiscards(75)
}
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
    "The particular variable to be sampled.

    Values between 0 and 22, point to MIB objects defined in
    IF-MIB [RFC2863].

    Values between 23 and 37, point to MIB objects defined in
    EtherLike-MIB [RFC2665].

    Values between 38 and 55, point to MIB objects defined in
    TOKENRING-MIB [RFC1748].

    Values between 56 and 72, point to MIB objects defined in
    RMON-MIB [RFC2819].

    Values between 73 and 75, point to MIB objects defined in
    BRIDGE-MIB [RFC1493].

    Because SNMP access control is articulated entirely in terms
    of the contents of MIB views, no access control mechanism
    exists that can restrict the value of this object to identify
    only those objects that exist in a particular MIB view.
    Because there is thus no acceptable means of restricting the
    read access that could be obtained through the TopN
    mechanism, the probe must only grant write access to this
    object in those views that have read access to all objects on
    the probe.
```

```
    During a set operation, if the supplied variable name is not
    available in the selected MIB view, or does not conform the
    other conditions mentioned above, a badValue error must be
    returned.

    This object may not be modified if the associated
    interfaceTopNControlStatus object is equal to active(1)."
 ::= { interfaceTopNControlEntry 2 }

interfaceTopNObjectSampleType OBJECT-TYPE
    SYNTAX INTEGER {
               absoluteValue(1),
               deltaValue(2),
               bandwidthPercentage(3)
           }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
     "The method of sampling the selected variable for storage in
     the interfaceTopNTable.

     If the value of this object is absoluteValue(1), the value of
     the selected variable will be copied directly into the topNValue.

     If the value of this object is deltaValue(2), the value of the
     selected variable at the last sample will be subtracted from
     the current value, and the difference will be stored in topNValue.

     If the value of this object is bandwidthPercentage(3), the agent
     records the total number of octets sent over an interval divided
     by the total number of octets that represent '100% bandwidth'
     for that interface. This ratio is multiplied by 1000 to
     retain a 3 digit integer (0..1000) in units of
     'tenth of one percent'. This type of computation is accurate for
     the octet counters. The usage of this option with respect to
     packets or error counters is not recommended.

     This object may not be modified if the associated
     interfaceTopNControlStatus object is equal to active(1)."
  ::= { interfaceTopNControlEntry 3 }


interfaceTopNNormalizationReq OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
    "This object indicates whether  normalization is required in the
```

```
    computation of the selected value.

    If the value of this object is 'true', the value of
    the selected variable will be multiplied by a factor equal to the
    interfaceTopNNormalizationFactor divided by the value of
    effective speed of the interface

    If the value of this object is 'false',
    the value of the selected variable will be taken 'as is' in
    the TopN computation.

    If the value of the object interfaceTopNSampleType is
    bandwidthPercentage(3), the object
    interfaceTopNNormalizationReq cannot take the value 'true'.

    The value of this object MUST be false if the effective speed of
    the interface sub-layer as determined from ifSpeed is zero. This
    conforms to the ifSpeed definition in [RFC2863]for a sub-layer
    that has no concept of bandwidth.

    This object may not be modified if the associated
    interfaceTopNControlStatus object is equal to active(1)."
  ::= { interfaceTopNControlEntry 4 }

interfaceTopNNormalizationFactor OBJECT-TYPE
    SYNTAX Integer32 (1..2147483647)
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
    "The value used for normalization if
    interfaceTopNNormalizationReq has the value 'true'.

 Example:
  The following set of values is applied to a device with multiple
  Ethernet interfaces running at 10 Mbps, 100 Mbps, and 1 Gbps.
  interfaceTopNObjectVariable = 'ifInOctets'
  interfaceTopNObjectSampleType = 'deltaValue'
  interfaceTopNNormalizationReq = 'true'
  interfaceTopNNormalizationFactor = 1000000000
  Applying this set of values will result in the sampled delta values
  to be multiplied by 100 for the 10 Mbps interfaces, and by 10 for
  the 100 Mbps interfaces, while the sample values for the 1 Gbps
  interface are left unchanged. The effective speed of the interface is
  taken from the value of ifSpeed for each interface, if ifSpeed is
  less than 4,294,967,295, or from ifHighSpeed multiplied by
  1,000,000 otherwise.

  At row creation the agent SHOULD set the value of this object to
```

```
  the effective speed of the interface."
::= { interfaceTopNControlEntry 5 }

interfaceTopNTimeRemaining OBJECT-TYPE
       SYNTAX      Integer32 (0..2147483647)
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
         "The number of seconds left in the report
         currently being collected.  When this object
         is modified by the management station, a new
         collection is started, possibly aborting a
         currently running report.  The new value is
         used as the requested duration of this report,
         which is loaded into the associated
         interfaceTopNDuration object.

         When this object is set to a non-zero value,
         any associated interfaceTopNEntries shall be
         made inaccessible by the agent.  While the value
         of this object is non-zero, it decrements by one
         per second until it reaches zero.  During this
         time, all associated interfaceTopNEntries shall
         remain inaccessible.  At the time that this object
         decrements to zero, the report is made accessible
         in the interfaceTopNTable.  Thus, the interfaceTopN
         table needs to be created only at the end of the
         collection interval.

         If the value of this object is set to zero
         while the associated report is running, the
         running report is aborted and no associated
         interfaceTopNEntries are created."
     DEFVAL { 0 }
     ::= { interfaceTopNControlEntry 6 }

   interfaceTopNDuration OBJECT-TYPE
       SYNTAX      Integer32 (0..2147483647)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
        "The number of seconds that this report has
        collected during the last sampling interval,
        or if this report is currently being collected,
        the number of seconds that this report is being
        collected during this sampling interval.

        When the associated interfaceTopNTimeRemaining
```

```
        object is set, this object shall be set by the
        agent to the same value and shall not be modified
        until the next time the interfaceTopNTimeRemaining
        is set.

        This value shall be zero if no reports have been
        requested for this interfaceTopNControlEntry."
      ::= { interfaceTopNControlEntry 7 }

   interfaceTopNRequestedSize OBJECT-TYPE
       SYNTAX      Integer32
       MAX-ACCESS  read-create
       STATUS      current
       DESCRIPTION
         "The maximum number of interfaces requested
         for the Top N Table.

         When this object is created or modified, the
         agent should set interfaceTopNGrantedSize as close
         to this object as is possible for the particular
         implementation and available resources."
       DEFVAL { 10 }
       ::= { interfaceTopNControlEntry 8 }

   interfaceTopNGrantedSize OBJECT-TYPE
       SYNTAX      Integer32 (0.. 2147483647)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
         "The maximum number of interfaces in the
         top N table.

         When the associated interfaceTopNRequestedSize object is
         created or modified, the agent should set this object as
         closely to the requested value as is possible for the
         particular implementation and available resources.  The
         agent must not lower this value except as a result of a
         set to the associated interfaceTopNRequestedSize object."
      ::= { interfaceTopNControlEntry 9 }

   interfaceTopNStartTime OBJECT-TYPE
       SYNTAX      TimeStamp
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
       "The value of sysUpTime when this top N report was
       last started.  In other words, this is the time that
       the associated interfaceTopNTimeRemaining object was
```

```
        modified to start the requested report.

        If the report has not yet been started, the value
        of this object is zero."
    ::= { interfaceTopNControlEntry 10 }

    interfaceTopNOwner OBJECT-TYPE
        SYNTAX      OwnerString
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
                "The entity that configured this entry and is
                using the resources assigned to it."
        ::= { interfaceTopNControlEntry 11 }

    interfaceTopNLastCompletionTime OBJECT-TYPE
        SYNTAX      TimeStamp
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
         "The value of sysUpTime when this top N report was
         last completed. If no report was yet completed, the value
         of this object is zero."
      ::= { interfaceTopNControlEntry 12 }


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

               If the value of this object is not equal to
               active(1), all associated entries in the
               interfaceTopNTable shall be deleted by the
               agent."
        ::= { interfaceTopNControlEntry 13 }


    -- Interface Top "N" reports

    interfaceTopNTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF InterfaceTopNEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "A table of reports for the top 'N' ports based on
```

```
          setting of associated control table entries. The
          maximum number of entries depends on the number
          of entries in table interfaceTopNControlTable and
          the value of object interfaceTopNGrantedSize for
          each entry.

          For each entry in the interfaceTopNControlTable,
          interfaces with the highest value of
          interfaceTopNValue shall be placed in this table
          in decreasing order of that rate until there is
          no more room or until there are no more ports."
        ::= { interfaceTopNObjects 3 }

    interfaceTopNEntry OBJECT-TYPE
        SYNTAX      InterfaceTopNEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
                "A set of statistics for an interface that is
                part of a top N report."
        INDEX    { interfaceTopNControlIndex,
                   interfaceTopNIndex }
        ::= { interfaceTopNTable 1 }

    InterfaceTopNEntry ::= SEQUENCE {
        interfaceTopNIndex
            Integer32,
        interfaceTopNDataSourceIndex
            Integer32,
        interfaceTopNValue
            Gauge32,
        interfaceTopNValue64
            CounterBasedGauge64
    }

    interfaceTopNIndex OBJECT-TYPE
        SYNTAX      Integer32 (1..65535)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
          "An index that uniquely identifies an entry in
          the interfaceTopN table among those in the same
          report.  This index is between 1 and N, where N
          is the number of entries in this report.  Increasing
          values of interfaceTopNIndex shall be assigned to
          entries with decreasing values of interfaceTopNValue
          or interfaceTopNValue64, whichever applies,
          until index N is assigned to the entry with the
```

```
          lowest value of interfaceTopNValue /
          interfaceTopNValue64 or there are no
          more interfaceTopNEntries.

          No ports are included in a report where their
          value of interfaceTopNValue would be zero."
      ::= { interfaceTopNEntry 1 }

   interfaceTopNDataSourceIndex OBJECT-TYPE
       SYNTAX      Integer32  (1..2147483647)
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
         "This object identifies the index corresponding
          to the dataSource for this entry.

          For sorted values of variables belonging to the
          IF-MIB, EtherLike-MIB or TOKENRING-MIB, this value
          equals the ifIndex of the interface.

          For sorted values of variables belonging to the
          RMON-MIB, this value equals the interface corresponding
          to the data source, pointed to by the value
          of etherStatsDataSource.

          For sorted values of variables belonging to the
          BRIDGE-MIB, this value equals the interface corresponding
          to the bridge port, pointed to by the value
          of dot1dBasePortIfIndex."
       ::= { interfaceTopNEntry 2 }

   interfaceTopNValue OBJECT-TYPE
       SYNTAX      Gauge32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
         "The value at the end of the sampling interval, or
         the amount of change in the selected variable
         during this sampling interval for the identified
         interface.  The selected variable is that interfaces's
         instance of the object selected by
         interfaceTopNObjectVariable. This value may be normalized
         if interfaceTopNNormalization required equals 'true'.
         This value of this object will be computed for all
         cases when interfaceTopNObjectVariable points to a
         32-bit counter or Gauge or when
         interfaceTopNObjectSampleType equals bandwidthPercentage(3),
         and will be zero for all other cases."
```

```
       ::= { interfaceTopNEntry 3 }

   interfaceTopNValue64 OBJECT-TYPE
       SYNTAX      CounterBasedGauge64
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
         "The value at the end of the sampling interval, or
         the amount of change in the selected variable
         during this sampling interval for the identified
         interface.  The selected variable is that interfaces's
         instance of the object selected by
         interfaceTopNObjectVariable. This value may be normalized
         if interfaceTopNNormalization required equals 'true'.
         This value of this object will be computed for all
         cases when interfaceTopNObjectVariable points to
         a 64-bit counter, and will be zero for all other cases."
       ::= { interfaceTopNEntry 4 }


--
-- Notifications Section
-- (none defined)
--

--
-- Conformance Section
--

interfaceTopNCompliances OBJECT IDENTIFIER ::=
                                          {interfaceTopNConformance 1 }
interfaceTopNGroups      OBJECT IDENTIFIER ::=
                                          {interfaceTopNConformance 2 }

interfaceTopNCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
            "Describes the requirements for conformance to the
            InterfaceTopN MIB."
        MODULE  -- this module
        MANDATORY-GROUPS { interfaceTopNGroup }
      ::= { interfaceTopNCompliances 1 }

interfaceTopNGroup OBJECT-GROUP
    OBJECTS {
       interfaceTopNCaps,
       interfaceTopNObjectVariable,
       interfaceTopNObjectSampleType,
```

```
       interfaceTopNNormalizationReq,
       interfaceTopNNormalizationFactor,
       interfaceTopNTimeRemaining,
       interfaceTopNDuration,
       interfaceTopNRequestedSize,
       interfaceTopNGrantedSize,
       interfaceTopNStartTime,
       interfaceTopNOwner,
       interfaceTopNLastCompletionTime,
       interfaceTopNRowStatus,
       interfaceTopNDataSourceIndex,
       interfaceTopNValue,
       interfaceTopNValue64
    }
    STATUS  current
    DESCRIPTION
         "A collection of objects providing interfaceTopN data for
         a multiple interfaces device."
::= { interfaceTopNGroups 1 }

END
```

7. References

[RFC2571] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.

[RFC1155] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.

[RFC1212] Rose, M., and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.

[RFC1215] M. Rose, "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.

[RFC2578] 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.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC1157] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.

[RFC1901] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.

[RFC1906] 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.

[RFC2572] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.

[RFC2574] Blumenthal, U., and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.

[RFC1905] 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.

[RFC2573] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", RFC 2573, April 1999.

[RFC2575] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.

[RFC2570] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2819] Waldbusser, S., "Remote Network Monitoring Management Information Base", STD 59, RFC 2819, May 2000.

[RFC2613] Waterman, R., Lahaye, B., Romascanu, D., and S. Waldbusser, "Remote Network Monitoring MIB Extensions for Switched Networks, Version 1.0", RFC 2613, June 1999.

[RFC1213] 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.

[RFC2863] McCloghrie, K., and Kastenholtz, F., "The Interfaces Group MIB", RFC 2863, June 2000.

[RFC2982] Kavasseri, R., Stewart B., "Distributed Management Expression MIB", RFC 2982, October 2000

[RFC2856] Bierman, A., McCloghrie, K., and Presuhn R., "Textual Conventions for Additional High Capacity Data Types", RFC 2856, June 2000.

[RFC2665] Flick, J., and Johnson, J., "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 2665, August 1999.

[RFC1748] McCloghrie, K., and Decker E., "IEEE802.5 MIB Using SMIv2", RFC 1748, December 1994.

[RFC1493] E. Decker, P. Langille, A. Rijsinghani, and McCloghrie, K., "Definition of Managed Objects for Bridges", RFC 1493, July 1993.

8. 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.

9. 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:

```
interfaceTopNDataSourceIndex
interfaceTopNValue
```

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 implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model [RFC2274] and the View-based Access Control Model [RFC2275] 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.

10. Author's Address

Dan Romascanu
Avaya Inc.
Atidim Technology Park, Bldg. #3
Tel Aviv, 61131
Israel

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

11. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.