IPv6 MIB Revision Design Team   Bill Fenner
INTERNET-DRAFT   AT&T Research
Expires: August 2001   Brian Haberman
                                                 Nortel Networks
                                                Keith McCloghrie
                                                   Cisco Systems
                                           Juergen Schoenwalder
                                                 TU Braunschweig
                                                    Dave Thaler
                                                      Microsoft
                                                  February 2001

Management Information Base
for the User Datagram Protocol (UDP)
draft-ops-rfc2013-update-00.txt

Status of this Document

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

Internet-Drafts are working documents of the Internet Engineering Task
Force (IETF), its areas, and its working groups. Note that other groups
may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months
and may be updated, replaced, or obsoleted by other documents at any
time. It is inappropriate to use Internet-Drafts as reference material
or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html.

This document is a product of the IPv6 MIB Revision Design Team.
Comments should be addressed to the authors, or the mailing list at
ipv6mib@ibr.cs.tu-bs.de.

Copyright Notice

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects used for implementations of the User Datagram Protocol (UDP) [4] in an IP version independent manner.

Table of Contents

1.   The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

o An overall architecture, described in RFC 2571 [5].

o 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 [6], STD 16, RFC 1212 [7] and RFC 1215 [8]. The second version, called SMIv2, is described in STD 58, RFC 2578 [9], STD 58, RFC 2579 [10] and STD 58, RFC 2580 [11].

o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [12]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [13] and RFC 1906 [14]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [14], RFC 2572 [15] and RFC 2574 [16].

o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [12]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [17].

o A set of fundamental applications described in RFC 2573 [18] and the
  view-based access control mechanism described in RFC 2575 [19].

A more detailed introduction to the current SNMP Management Framework
can be found in RFC 2570 [20].

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.

2. Revision History

Changes from first draft posted to v6mib mailing list:

23 Feb 2001

    Made threshold for HC packet counters 1Mpps

    Added copyright statements and table of contents

21 Feb 2001 -- Juergen's changes

    Renamed udpInetTable to udpListenerTable

    Updated Conformance info

6 Feb 2001

    Removed v6-only objects.

    Removed remote and instance objects, turning the table back into a
    listener-only table.

    Renamed inetUdp* to udpInet*

    Added HC in and out datagram counters

      Added SIZE restriction to udpListenerLocalAddress. (36 = 32-byte
      addresses plus 4-byte scope, but it's just a strawman)

      Used InetPortNumber TC from updated INET-ADDRESS-MIB

      Updated compliance statements.

      Added Keith to authors

      Added open issues section.

[3]. Definitions

```
UDP-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Counter32, Counter64,
    IpAddress, mib-2                   FROM SNMPv2-SMI
    MODULE-COMPLIANCE, OBJECT-GROUP    FROM SNMPv2-CONF
    InetAddress, InetAddressType,
    InetPortNumber                     FROM INET-ADDRESS-MIB;


udpMIB MODULE-IDENTITY
    LAST-UPDATED "200102210000Z"
    ORGANIZATION "IETF IPv6 MIB Revision Team"
    CONTACT-INFO
            "Bill Fenner (editor)

             AT&T Labs -- Research
             75 Willow Rd.
             Menlo Park, CA 94025

             Phone: +1 650 330-7893
             Email: <fenner@research.att.com>"
    DESCRIPTION
            "The MIB module for managing UDP implementations."
    REVISION      "200102210000Z"
    DESCRIPTION
            "IP version neutral revision, published as RFC XXXX."
    REVISION      "9411010000Z"
    DESCRIPTION
            "Initial SMIv2 version, published as RFC 2013."
    REVISION      "9103310000Z"
    DESCRIPTION
            "The initial revision of this MIB module was part of MIB-II."
    ::= { mib-2 50 }
```

```
-- the UDP group

udp       OBJECT IDENTIFIER ::= { mib-2 7 }

udpInDatagrams OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
          "The total number of UDP datagrams delivered to UDP users."
    ::= { udp 1 }

udpNoPorts OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
          "The total number of received UDP datagrams for which there
           was no application at the destination port."
    ::= { udp 2 }

udpInErrors OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
          "The number of received UDP datagrams that could not be
           delivered for reasons other than the lack of an application
           at the destination port."
    ::= { udp 3 }

udpOutDatagrams OBJECT-TYPE
    SYNTAX     Counter32
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
          "The total number of UDP datagrams sent from this entity."
    ::= { udp 4 }

udpHCInDatagrams OBJECT-TYPE
    SYNTAX     Counter64
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
          "The total number of UDP datagrams delivered to UDP users,
           for devices which can receive more than 1 million UDP
           packets per second."
    ::= { udp 26 }
```

udpHCOutDatagrams OBJECT-TYPE
    SYNTAX   Counter64
    MAX-ACCESS read-only
    STATUS   current
    DESCRIPTION
            "The total number of UDP datagrams sent from this entity, for
             devices which can transmit more than 1 million UDP packets
             per second."
    ::= { udp 27 }

-- The UDP Listener table

-- The UDP listener table contains information about this
-- entity's UDP end-points on which a local application is
-- currently accepting datagrams.

udpListenerTable OBJECT-TYPE
    SYNTAX   SEQUENCE OF UdpListenerEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
            "A table containing UDP listener information."
    ::= { udp 7 }

udpListenerEntry OBJECT-TYPE
    SYNTAX   UdpListenerEntry
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
            "Information about a particular current UDP listener."
    INDEX   { udpListenerLocalAddressType,
              udpListenerLocalAddress,
              udpListenerLocalPort }
    ::= { udpListenerTable 1 }

UdpListenerEntry ::= SEQUENCE {
        udpListenerLocalAddressType InetAddressType,
        udpListenerLocalAddress   InetAddress,
        udpListenerLocalPort   InetPortNumber
    }

udpListenerLocalAddressType OBJECT-TYPE
    SYNTAX   InetAddressType
    MAX-ACCESS not-accessible
    STATUS   current
    DESCRIPTION
            "The address type of udpListenerLocalAddress"

```
     ::= { udpListenerEntry 1 }

udpListenerLocalAddress OBJECT-TYPE
    SYNTAX     InetAddress (SIZE(0..36))
    MAX-ACCESS not-accessible
    STATUS     current
    DESCRIPTION
           "The local IP address for this UDP listener.  In the case of
            a UDP listener which is willing to accept datagrams for any
            IP interface associated with the node, a value of all zeroes
            is used."
    ::= { udpListenerEntry 2 }

udpListenerLocalPort OBJECT-TYPE
    SYNTAX     InetPortNumber
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
           "The local port number for this UDP listener."
    ::= { udpListenerEntry 3 }



-- The deprecated UDP Listener table

-- The UDP listener table contains information about this
-- entity's IPv4 UDP end-points on which a local application is
-- currently accepting datagrams.

udpTable OBJECT-TYPE
    SYNTAX     SEQUENCE OF UdpEntry
    MAX-ACCESS not-accessible
    STATUS     deprecated
    DESCRIPTION
           "A table containing IPv4-specific UDP listener information.
            It contains information about all local IPv4 UDP end-points
            on which an application is currently accepting datagrams.
            This table has been deprecated in favor of the version
            neutral udpListenerTable."
    ::= { udp 5 }

udpEntry OBJECT-TYPE
    SYNTAX     UdpEntry
    MAX-ACCESS not-accessible
    STATUS     deprecated
    DESCRIPTION
           "Information about a particular current UDP listener."
    INDEX   { udpLocalAddress, udpLocalPort }
```

    ::= { udpTable 1 }

UdpEntry ::= SEQUENCE {
        udpLocalAddress IpAddress,
        udpLocalPort   INTEGER
    }

udpLocalAddress OBJECT-TYPE
    SYNTAX   IpAddress
    MAX-ACCESS read-only
    STATUS   deprecated
    DESCRIPTION
            "The local IP address for this UDP listener. In the case of
             a UDP listener which is willing to accept datagrams for any
             IP interface associated with the node, the value 0.0.0.0 is
             used."
    ::= { udpEntry 1 }

udpLocalPort OBJECT-TYPE
    SYNTAX   INTEGER (0..65535)
    MAX-ACCESS read-only
    STATUS   deprecated
    DESCRIPTION
            "The local port number for this UDP listener."
    ::= { udpEntry 2 }

-- conformance information

udpMIBConformance OBJECT IDENTIFIER ::= { udpMIB 2 }

udpMIBCompliances OBJECT IDENTIFIER ::= { udpMIBConformance 1 }
udpMIBGroups   OBJECT IDENTIFIER ::= { udpMIBConformance 2 }

-- compliance statements

udpMIBCompliance2 MODULE-COMPLIANCE
    STATUS   current
    DESCRIPTION
            "The compliance statement for systems which implement UDP."
    MODULE -- this module
        MANDATORY-GROUPS { udpBaseGroup, udpListenerGroup }
        GROUP   udpHCGroup
        DESCRIPTION
            "This group is mandatory for those systems which are capable
                of receiving or transmitting more than 1 million UDP

```
                  packets per second.  1 million packets per second will
                  cause a Counter32 to wrap in just over an hour."
        ::= { udpMIBCompliances 2 }


udpMIBCompliance MODULE-COMPLIANCE
        STATUS      deprecated
        DESCRIPTION
              "The compliance statement for IPv4-only systems which
               implement UDP.  For IP version independence, this compliance
               statement is deprecated in favor of udpMIBCompliance2."
        MODULE  -- this module
            MANDATORY-GROUPS { udpGroup }
        ::= { udpMIBCompliances 1 }

-- units of conformance

udpGroup OBJECT-GROUP
        OBJECTS   { udpInDatagrams, udpNoPorts,
                     udpInErrors, udpOutDatagrams,
                     udpLocalAddress, udpLocalPort }
        STATUS      deprecated
        DESCRIPTION
              "The udp group of objects providing for management of UDP
               over IPv4."
        ::= { udpMIBGroups 1 }


udpBaseGroup OBJECT-GROUP
        OBJECTS   { udpInDatagrams, udpNoPorts, udpInErrors, udpOutDatagrams }
        STATUS      current
        DESCRIPTION
              "The group of objects providing for counters of UDP
               statistics."
        ::= { udpMIBGroups 2 }


udpHCGroup OBJECT-GROUP
        OBJECTS   { udpHCInDatagrams, udpHCOutDatagrams }
        STATUS      current
        DESCRIPTION
              "The group of objects providing for counters of high speed
               UDP implementations."
        ::= { udpMIBGroups 3 }


udpListenerGroup OBJECT-GROUP
        OBJECTS   { udpListenerLocalPort }
        STATUS      current
        DESCRIPTION
              "The group of objects providing for the IP version
               independent management of UDP listeners."
```

```
    ::= { udpMIBGroups 4 }
```

END

4. Open Issues

[optional] connection table to more fully specify sockets?

Per-connection/listener datagram / octet count objects in an optional
conformance group?

5. Acknowledgements

This document contains a modified subset of RFC 1213 and updates RFC
2013 and RFC 2454.

6. References

[1] Rose, M. and K. McCloghrie, "Management Information Base for Network
     Management of TCP/IP-based internets", RFC 1213, March 1991.

[2] K. McCloghrie, "SNMPv2 Management Information Base for the User
     Datagram Protocol using SMIv2", RFC 2013, November 1996.

[3] Haskin, D. and S. Onishi, "IP Version 6 Management Information Base
     for the User Datagram Protocol", RFC 2454, December 1998.

[4] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DARPA, August
     1980.

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

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

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

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

[9] 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.

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

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

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

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

[14] 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.

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

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

[17] 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.

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

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

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

7. Security Considerations

There are no management objects defined in this MIB that have a MAX-
ACCESS clause of read-write and/or read-create. So, if this MIB is
implemented correctly, then there is no risk that an intruder can alter
or create any management objects of this MIB via direct SNMP SET
operations.

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

o The udpListenerLocalPort and udpLocalPort objects can be used to
  identify what ports are open on the machine and can thus what attacks
  are likely to succeed, without the attacker having to run a port
  scanner.

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 RFC 2574 [16] and the View-based Access Control
Model RFC 2575 [19] 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.

8. Editor's Address

Bill Fenner
AT&T Labs -- Research
[75](#) Willow Rd
Menlo Park, CA 94025
USA

Email: fenner@research.att.com

[9](#). Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

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

The limited permissions granted above are perpetual and will not be
revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS
IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK
FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT
LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT
INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR
FITNESS FOR A PARTICULAR PURPOSE.