                     Definitions of Managed Objects for
                          Source Routing Bridges

Status of this Memo

   This RFC 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" for the standardization state and status
   of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Introduction

   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 source routing and
   source routing transparent bridges. These bridges are also required
   to implement relevant groups in the Bridge MIB [6].

   This MIB supersedes the dot1dSr group of objects published in an
   earlier version of the Bridge MIB, RFC 1286. Changes have primarily
   been made to track changes in the IEEE 802.5M SRT Addendum to the
   IEEE 802.1D Standard for MAC Bridges.

2. The Network Management Framework

   The Internet-standard Network Management Framework consists of three
   components. They are:

      o   STD 16, RFC 1155 which defines the SMI, the mechanisms used for
            describing and naming objects for the purpose of
            management. STD 16, RFC 1212 defines a more concise description
            mechanism, which is wholly consistent with the SMI.

      o   STD 17, RFC 1213 defines MIB-II, the core set of managed objects
            for the Internet suite of protocols.

      o   STD 15, RFC 1157 which defines the SNMP, the protocol used for
            network access to managed objects.

   The Framework permits new objects to be defined for the purpose of
   experimentation and evaluation.

2.1. Object Definitions

   Managed objects are accessed via a virtual information store, termed
   the Management Information Base or MIB. Objects in the MIB are
   defined using the subset of Abstract Syntax Notation One (ASN.1)
   defined in the SMI. In particular, each object object type is named
   by an OBJECT IDENTIFIER, an administratively assigned name. The
   object type together with an object instance serves to uniquely
   identify a specific instantiation of the object. For human
   convenience, we often use a textual string, termed the descriptor, to
   refer to the object type.

3. Overview

   A common device present in many networks is the Bridge. This device
   is used to connect Local Area Network segments below the network

layer. There are two major modes defined for this bridging;
transparent and source route. The transparent method of bridging is
defined in the IEEE 802.1d MAC Bridge specification [11]. Source
route bridging has been defined by I.B.M. and is described in the
Token Ring Architecture Reference [12], as well as the IEEE 802.5M
SRT Bridge Operations Addendum [14] to 802.1d. This memo defines
objects needed for management of a source routing bridge, and is an
extension to the SNMP Bridge MIB [6].

An explicit attempt was made to keep this MIB as simple as possible.
This was accomplished by applying the following criteria to objects
proposed for inclusion:

   (1)   Start with a small set of essential objects and add only
         as further objects are needed.

   (2)   Require objects be essential for either fault or
         configuration management.

   (3)   Consider evidence of current use and/or utility.

   (4)   Limit the total of objects.

   (5)   Exclude objects which are simply derivable from others in
         this or other MIBs.

   (6)   Avoid causing critical sections to be heavily
         instrumented. The guideline that was followed is one
         counter per critical section per layer.

3.1. Structure of MIB

   Objects in this MIB are arranged into groups. Each group is
   organized as a set of related objects. The overall structure and
   assignment of objects to their groups is shown below. Where
   appropriate, the corresponding management object name found in IEEE
   802.1d [11] and IEEE 802.5M [14] is also included.

    SR Bridge MIB Name   IEEE Name

       dot1dSr
         PortTable
           Port
           HopCount   SourceRoutingPort
                                         .PortHopCount
           LocalSegment   .SegmentNumber
           BridgeNum   .BridgeNumber
           TargetSegment

```
          LargestFrame                    .LargestFrameSize
          STESpanMode                     .LimitedBroadcastMode
          SpecInFrames                    BridgePort
                                            .ValidSRFramesReceived
          SpecOutFrames                   .ValidSRForwardedOutbound
          ApeInFrames
          ApeOutFrames                    .BroadcastFramesForwarded
          SteInFrames
          SteOutFrames                    .BroadcastFramesForwarded
          SegmentMismatchDiscards         .DiscardInvalidRI
          DuplicateSegmentDiscards        .LanIdMismatch
          HopCountExceededDiscards        .FramesDiscardedHopCountExceeded
```

The following IEEE management objects have not been included in the
SR Bridge MIB for the indicated reasons.

```
  IEEE Object                     Disposition

  SourceRoutingPort

                                  The following objects were NOT
                                  included in this MIB because they
                                  are redundant or not considered
                                  useful.
        .LimitedBroadcastEnable
        .DiscardLackOfBuffers
        .DiscardErrorDetails
        .DiscardTargetLANInoperable
        .ValidSRDiscardedInbound
        .BroadcastBytesForwarded
        .NonBroadcastBytesForwarded
        .FramesNotReceivedDueToCongestion
        .FramesDiscardedDueToInternalError
```

3.1.1. The dot1dSr Group

   This group contains the objects that describe the entity's state with
   respect to source route bridging. If source routing is not
   supported, this group will not be implemented. This group is
   applicable to source route only, and SRT bridges.

3.1.2. The dot1dPortPair Group

   Implementation of this group is optional. This group is implemented
   by those bridges that support the port-pair multiport model of the
   source route bridging mode as defined in the IEEE 802.5M SRT Addendum
   to 802.1d.

3.2. Relationship to Other MIBs

   As described above, some IEEE 802.1d management objects have not been
   included in this MIB because they overlap with objects in other MIBs
   applicable to a bridge implementing this MIB. In particular, it is
   assumed that a bridge implementing this MIB will also implement (at
   least) the Bridge MIB and the 'system' group and the 'interfaces'
   group defined in MIB-II [4].

3.2.1. Relationship to the Bridge MIB

   The Bridge MIB [6] must be implemented by all bridges, including
   transparent, SR and SRT bridges. The SR bridge MIB is an extension
   to the Bridge MIB.

3.2.2. Relationship to the 'system' group

   In MIB-II, the 'system' group is defined as being mandatory for all
   systems such that each managed entity contains one instance of each
   object in the 'system' group. Thus, those objects apply to the
   entity as a whole irrespective of whether the entity's sole
   functionality is bridging, or whether bridging is only a subset of
   the entity's functionality.

3.2.3. Relationship to the 'interfaces' group

   In MIB-II, the 'interfaces' group is defined as being mandatory for
   all systems and contains information on an entity's interfaces, where
   each interface is thought of as being attached to a 'subnetwork'.
   (Note that this term is not to be confused with 'subnet' which refers
   to an addressing partitioning scheme used in the Internet suite of
   protocols.) The term 'segment' is used in this memo to refer to such
   a subnetwork.

   Implicit in this MIB is the notion of ports on a bridge. Each of
   these ports is associated with one interface of the 'interfaces'
   group, and in most situations, each port is associated with a
   different interface. However, there are situations in which multiple
   ports are associated with the same interface. An example of such a
   situation would be several ports, each corresponding one-to-one with
   several X.25 virtual circuits, but all on the same interface.

   Each port is uniquely identified by a port number. A port number has
   no mandatory relationship to an interface number, but in the simple
   case, a port number will have the same value as the corresponding
   interface's interface number.

   Some entities provide other services in addition to bridging with
   respect to the data sent and received by their interfaces. In such
   situations, only a subset of the data sent/received on an interface
   is within the domain of the entity's bridging functionality. This
   subset is considered to be delineated according to a set of
   protocols, with some protocols being bridged, and other protocols not
   being bridged. For example, in an entity which exclusively performed
   bridging, all protocols would be considered as being bridged, whereas
   in an entity which performed IP routing on IP datagrams and only
   bridged other protocols, only the non-IP data would be considered as
   being bridged.

   Thus, this MIB (and in particular, its counters) are applicable only
   to that subset of the data on an entity's interfaces which is
   sent/received for a protocol being bridged. All such data is
   sent/received via the ports of the bridge.

4. Changes from RFC 1286

   In addition to being separated from the Bridge MIB into a separate
   document, the following changes were implemented as a result of
   feedback from IEEE 802.5M:

        (1) Changed syntax of dot1dSrPortLargestFrame to INTEGER in
             order to allow for having 64 possible values as described
             in draft 7 of the SR Addendum. Listed all legal values
             in description.

        (2) Updated syntax of dot1dSrPort, used to index into
             dot1dSrPortTable, to use the range (1..65535).

        (3) Added a counter to dot1dSrPortTable to count occurrences
             of duplicate LAN IDs or Tree errors.

        (4) Added a counter to dot1dSrPortTable to count LAN ID
             mismatches.

        (5) Added text to dot1dSrPortSpecInFrames and
             dot1dSrPortSpecOutFrames clarifying that they are also
             referred to as Source Routed Frames.

        (6) Added text to dot1dSrPortApeInFrames and
             dot1dSrPortApeOutFrames clarifying that they are also
             referred to as All Routes Explorer frames.

        (7) Added a scalar variable to the dot1dSr group to indicate
             whether the bridge uses 3 bit or 6 bit length negotiation
             fields.

        (8) Added dot1dPortPairGroup to allow representation of port
             pairs as defined in the IEEE 802.5M SRT Addendum.

5. Definitions

        SOURCE-ROUTING-MIB DEFINITIONS ::= BEGIN

        IMPORTS
                Counter, Gauge
                        FROM RFC1155-SMI
                dot1dBridge, dot1dSr
                        FROM BRIDGE-MIB
                OBJECT-TYPE
                        FROM RFC-1212;

        -- groups in the SR MIB

        -- dot1dSr is imported from the Bridge MIB

        dot1dPortPair   OBJECT IDENTIFIER ::= { dot1dBridge 10 }

        -- the dot1dSr group

        -- this group is implemented by those bridges that
        -- support the source route bridging mode, including Source
        -- Routing and SRT bridges.

        dot1dSrPortTable OBJECT-TYPE
            SYNTAX   SEQUENCE OF Dot1dSrPortEntry
            ACCESS   not-accessible
            STATUS   mandatory
            DESCRIPTION
                    "A table that contains information about every
                    port that is associated with this source route
                    bridge."
            ::= { dot1dSr 1 }

        dot1dSrPortEntry OBJECT-TYPE
            SYNTAX   Dot1dSrPortEntry
            ACCESS   not-accessible
            STATUS   mandatory
            DESCRIPTION
                    "A list of information for each port of a source
                    route bridge."
            INDEX   { dot1dSrPort }

                    ::= { dot1dSrPortTable 1 }

            Dot1dSrPortEntry ::=
                SEQUENCE {
                    dot1dSrPort
                        INTEGER,
                    dot1dSrPortHopCount
                        INTEGER,
                    dot1dSrPortLocalSegment
                        INTEGER,
                    dot1dSrPortBridgeNum
                        INTEGER,
                    dot1dSrPortTargetSegment
                        INTEGER,
                    dot1dSrPortLargestFrame
                        INTEGER,
                    dot1dSrPortSTESpanMode
                        INTEGER,
                    dot1dSrPortSpecInFrames
                        Counter,
                    dot1dSrPortSpecOutFrames
                        Counter,
                    dot1dSrPortApeInFrames
                        Counter,
                    dot1dSrPortApeOutFrames
                        Counter,
                    dot1dSrPortSteInFrames
                        Counter,
                    dot1dSrPortSteOutFrames
                        Counter,
                    dot1dSrPortSegmentMismatchDiscards
                        Counter,
                    dot1dSrPortDuplicateSegmentDiscards
                        Counter,
                    dot1dSrPortHopCountExceededDiscards
                        Counter,
                    dot1dSrPortDupLanIdOrTreeErrors
                        Counter,
                    dot1dSrPortLanIdMismatches
                        Counter
                }

            dot1dSrPort OBJECT-TYPE
                SYNTAX   INTEGER (1..65535)
                ACCESS   read-only
                STATUS   mandatory
                DESCRIPTION
                        "The port number of the port for which this entry

                    contains Source Route management information."
            ::= { dot1dSrPortEntry 1 }

        dot1dSrPortHopCount OBJECT-TYPE
            SYNTAX INTEGER
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                    "The maximum number of routing descriptors allowed
                    in an All Paths or Spanning Tree Explorer frames."
            ::= { dot1dSrPortEntry 2 }

        dot1dSrPortLocalSegment OBJECT-TYPE
            SYNTAX INTEGER
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                    "The segment number that uniquely identifies the
                    segment to which this port is connected. Current
                    source routing protocols limit this value to the
                    range: 0 through 4095. (The value 0 is used by
                    some management applications for special test
                    cases.) A value of 65535 signifies that no segment
                    number is assigned to this port."
            ::= { dot1dSrPortEntry 3 }

        dot1dSrPortBridgeNum OBJECT-TYPE
            SYNTAX INTEGER
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                    "A bridge number uniquely identifies a bridge when
                    more than one bridge is used to span the same two
                    segments. Current source routing protocols limit
                    this value to the range: 0 through 15. A value of
                    65535 signifies that no bridge number is assigned
                    to this bridge."
            ::= { dot1dSrPortEntry 4 }

        dot1dSrPortTargetSegment OBJECT-TYPE
            SYNTAX INTEGER
            ACCESS read-write
            STATUS mandatory
            DESCRIPTION
                    "The segment number that corresponds to the target
                    segment this port is considered to be connected to
                    by the bridge. Current source routing protocols
                    limit this value to the range: 0 through 4095.

                        (The value 0 is used by some management
                        applications for special test cases.) A value of
                        65535 signifies that no target segment is assigned
                        to this port."
                ::= { dot1dSrPortEntry 5 }

        -- It would be nice if we could use ifMtu as the size of the
        -- largest frame, but we can't because ifMtu is defined to be
        -- the size that the (inter-)network layer can use which can
        -- differ from the MAC layer (especially if several layers of
        -- encapsulation are used).

        dot1dSrPortLargestFrame OBJECT-TYPE
            SYNTAX   INTEGER
            ACCESS   read-write
            STATUS   mandatory
            DESCRIPTION
                    "The maximum size of the INFO field (LLC and
                    above) that this port can send/receive. It does
                    not include any MAC level (framing) octets. The
                    value of this object is used by this bridge to
                    determine whether a modification of the
                    LargestFrame (LF, see [14]) field of the Routing
                    Control field of the Routing Information Field is
                    necessary.

                    64 valid values are defined by the IEEE 802.5M SRT
                    Addendum: 516, 635, 754, 873, 993, 1112, 1231,
                    1350, 1470, 1542, 1615, 1688, 1761, 1833, 1906,
                    1979, 2052, 2345, 2638, 2932, 3225, 3518, 3812,
                    4105, 4399, 4865, 5331, 5798, 6264, 6730, 7197,
                    7663, 8130, 8539, 8949, 9358, 9768, 10178, 10587,
                    10997, 11407, 12199, 12992, 13785, 14578, 15370,
                    16163, 16956, 17749, 20730, 23711, 26693, 29674,
                    32655, 35637, 38618, 41600, 44591, 47583, 50575,
                    53567, 56559, 59551, and 65535.

                    An illegal value will not be accepted by the
                    bridge."
                ::= { dot1dSrPortEntry 6 }

        dot1dSrPortSTESpanMode OBJECT-TYPE
            SYNTAX   INTEGER {
                    auto-span(1),
                    disabled(2),
                    forced(3)
                }
            ACCESS   read-write

```
                    STATUS  mandatory
                    DESCRIPTION
                            "Determines how this port behaves when presented
                            with a Spanning Tree Explorer frame.  The value
                            'disabled(2)' indicates that the port will not
                            accept or send Spanning Tree Explorer packets; any
                            STE packets received will be silently discarded.
                            The value 'forced(3)' indicates the port will
                            always accept and propagate Spanning Tree Explorer
                            frames.  This allows a manually configured
                            Spanning Tree for this class of packet to be
                            configured.  Note that unlike transparent
                            bridging, this is not catastrophic to the network
                            if there are loops.  The value 'auto-span(1)' can
                            only be returned by a bridge that both implements
                            the Spanning Tree Protocol and has use of the
                            protocol enabled on this port. The behavior of the
                            port for Spanning Tree Explorer frames is
                            determined by the state of dot1dStpPortState.  If
                            the port is in the 'forwarding' state, the frame
                            will be accepted or propagated.  Otherwise, it
                            will be silently discarded."
                    ::= { dot1dSrPortEntry 7 }

            dot1dSrPortSpecInFrames OBJECT-TYPE
                SYNTAX  Counter
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The number of Specifically Routed frames, also
                        referred to as Source Routed Frames, that have
                        been received from this port's segment."
                ::= { dot1dSrPortEntry 8 }

            dot1dSrPortSpecOutFrames OBJECT-TYPE
                SYNTAX  Counter
                ACCESS  read-only
                STATUS  mandatory
                DESCRIPTION
                        "The number of Specifically Routed frames, also
                        referred to as Source Routed Frames, that this
                        port has transmitted on its segment."
                ::= { dot1dSrPortEntry 9 }

            dot1dSrPortApeInFrames OBJECT-TYPE
                SYNTAX  Counter
                ACCESS  read-only
                STATUS  mandatory
```

            DESCRIPTION
                    "The number of All Paths Explorer frames, also
                    referred to as All Routes Explorer frames, that
                    have been received by this port from its segment."
            ::= { dot1dSrPortEntry 10 }

        dot1dSrPortApeOutFrames OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                    "The number of all Paths Explorer Frames, also
                    referred to as All Routes Explorer frames, that
                    have been transmitted by this port on its
                    segment."
            ::= { dot1dSrPortEntry 11 }

        dot1dSrPortSteInFrames OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                    "The number of spanning tree explorer frames that
                    have been received by this port from its segment."
            ::= { dot1dSrPortEntry 12 }

        dot1dSrPortSteOutFrames OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                    "The number of spanning tree explorer frames that
                    have been transmitted by this port on its
                    segment."
            ::= { dot1dSrPortEntry 13 }

        dot1dSrPortSegmentMismatchDiscards OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                    "The number of explorer frames that have been
                    discarded by this port because the routing
                    descriptor field contained an invalid adjacent
                    segment value."
            ::= { dot1dSrPortEntry 14 }

        dot1dSrPortDuplicateSegmentDiscards OBJECT-TYPE

```
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of frames that have been discarded by
                this port because the routing descriptor field
                contained a duplicate segment identifier."
        ::= { dot1dSrPortEntry 15 }

    dot1dSrPortHopCountExceededDiscards OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of explorer frames that have been
                discarded by this port because the Routing
                Information Field has exceeded the maximum route
                descriptor length."
        ::= { dot1dSrPortEntry 16 }

    dot1dSrPortDupLanIdOrTreeErrors OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of duplicate LAN IDs or Tree errors.
                This helps in detection of problems in networks
                containing older IBM Source Routing Bridges."
        ::= { dot1dSrPortEntry 17 }

    dot1dSrPortLanIdMismatches OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The number of ARE and STE frames that were
                discarded because the last LAN ID in the routing
                information field did not equal the LAN-in ID.
                This error can occur in implementations which do
                only a LAN-in ID and Bridge Number check instead
                of a LAN-in ID, Bridge Number, and LAN-out ID
                check before they forward broadcast frames."
        ::= { dot1dSrPortEntry 18 }


    -- scalar object in dot1dSr

    dot1dSrBridgeLfMode OBJECT-TYPE
```

```
                  SYNTAX  INTEGER {
                          mode3(1),
                          mode6(2)
                      }
              ACCESS  read-write
              STATUS  mandatory
              DESCRIPTION
                      "Indicates whether the bridge operates using older
                      3 bit length negotiation fields or the newer 6 bit
                      length field in its RIF."
              ::= { dot1dSr 2 }


      -- The Port-Pair Database

      -- Implementation of this group is optional.

      -- This group is implemented by those bridges that support
      -- the direct multiport model of the source route bridging
      -- mode as defined in the IEEE 802.5 SRT Addendum to
      -- 802.1d.

      -- Bridges implementing this group may report 65535 for
      -- dot1dSrPortBridgeNumber and dot1dSrPortTargetSegment,
      -- indicating that those objects are not applicable.

      dot1dPortPairTableSize OBJECT-TYPE
          SYNTAX  Gauge
          ACCESS  read-only
          STATUS  mandatory
          DESCRIPTION
                  "The total number of entries in the Bridge Port
                  Pair Database."
           ::= { dot1dPortPair 1 }


      -- the Bridge Port-Pair table

      -- this table represents port pairs within a bridge forming
      -- a unique bridge path, as defined in the IEEE 802.5M SRT
      -- Addendum.

      dot1dPortPairTable OBJECT-TYPE
          SYNTAX  SEQUENCE OF Dot1dPortPairEntry
          ACCESS  not-accessible
          STATUS  mandatory
          DESCRIPTION
                  "A table that contains information about every
```

```
                        port pair database entity associated with this
                        source routing bridge."
                ::= { dot1dPortPair 2 }

        dot1dPortPairEntry OBJECT-TYPE
            SYNTAX  Dot1dPortPairEntry
            ACCESS  not-accessible
            STATUS  mandatory
            DESCRIPTION
                    "A list of information for each port pair entity
                    of a bridge."
            INDEX   { dot1dPortPairLowPort, dot1dPortPairHighPort }
            ::= { dot1dPortPairTable 1 }

        Dot1dPortPairEntry ::=
            SEQUENCE {
                dot1dPortPairLowPort
                    INTEGER,
                dot1dPortPairHighPort
                    INTEGER,
                dot1dPortPairBridgeNum
                    INTEGER,
                dot1dPortPairBridgeState
                    INTEGER
            }

        dot1dPortPairLowPort OBJECT-TYPE
            SYNTAX  INTEGER (1..65535)
            ACCESS  read-write
            STATUS  mandatory
            DESCRIPTION
                    "The port number of the lower numbered port for
                    which this entry contains port pair database
                    information."
            ::= { dot1dPortPairEntry 1 }

        dot1dPortPairHighPort OBJECT-TYPE
            SYNTAX  INTEGER (1..65535)
            ACCESS  read-write
            STATUS  mandatory
            DESCRIPTION
                    "The port number of the higher numbered port for
                    which this entry contains port pair database
                    information."
            ::= { dot1dPortPairEntry 2 }

        dot1dPortPairBridgeNum OBJECT-TYPE
            SYNTAX  INTEGER
```

```
                 ACCESS   read-write
                 STATUS   mandatory
                 DESCRIPTION
                         "A bridge number that uniquely identifies the path
                         provided by this source routing bridge between the
                         segments connected to dot1dPortPairLowPort and
                         dot1dPortPairHighPort.  The purpose of bridge
                         number is to disambiguate between multiple paths
                         connecting the same two LANs."
                 ::= { dot1dPortPairEntry 3 }

           dot1dPortPairBridgeState OBJECT-TYPE
                 SYNTAX   INTEGER {
                         enabled(1),
                         disabled(2),
                         invalid(3)
                     }
                 ACCESS   read-write
                 STATUS   mandatory
                 DESCRIPTION
                         "The state of dot1dPortPairBridgeNum.  Writing
                         'invalid(3)' to this object removes the
                         corresponding entry."
                 ::= { dot1dPortPairEntry 4 }



           END
```

6. Acknowledgments

   This document was produced on behalf of the Bridge MIB Working Group
   in the NM area of the Internet Engineering Task Force.

   The authors wish to thank the members of the Bridge MIB Working Group
   for their many comments and suggestions which improved this effort.

7. References

   [1] Cerf, V., "IAB Recommendations for the Development of Internet
       Network Management Standards", RFC 1052, NRI, April 1988.

   [2] Cerf, V., "Report of the Second Ad Hoc Network Management Review
       Group", RFC 1109, NRI, August 1989.

   [3] Rose M., and K. McCloghrie, "Structure and Identification of
       Management Information for TCP/IP-based internets", STD 16, RFC

          1155, Performance Systems International, Hughes LAN Systems, May
          1990.

     [4] McCloghrie K., and M. Rose, Editors, "Management Information Base
         for Network Management of TCP/IP-based internets", STD 17, RFC
         1213, Performance Systems International, March 1991.

     [5] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple
         Network Management Protocol", STD 15, RFC 1157, SNMP Research,
         Performance Systems International, Performance Systems
         International, MIT Laboratory for Computer Science, May 1990.

     [6] Decker, E., Langille, P., Rijsinghani, A., and McCloghrie, K.,
         "Definitions of Managed Objects for Bridges", RFC 1493, cisco
         Systems, Digital Equipment Corporation, Digital Equipment
         Corporation, Hughes LAN Systems, July 1993.

     [7] Information processing systems - Open Systems Interconnection -
         Specification of Abstract Syntax Notation One (ASN.1),
         International Organization for Standardization, International
         Standard 8824, December 1987.

     [8] Information processing systems - Open Systems Interconnection -
         Specification of Basic Encoding Rules for Abstract Notation One
         (ASN.1), International Organization for Standardization,
         International Standard 8825, December 1987.

     [9] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions",
         STD 16, RFC 1212, Performance Systems International, Hughes LAN
         Systems, March 1991.

    [10] Rose, M., Editor, "A Convention for Defining Traps for use with
         the SNMP", RFC 1215, Performance Systems International, March
         1991.

    [11] ANSI/IEEE Standard 802.1D-1990 MAC Bridges, IEEE Project 802
         Local and Metropolitan Area Networks, (March 8, 1991).

    [12] I.B.M. Token Ring Architecture Reference.

    [13] ISO DIS 10038 MAC Bridges.

    [14] ANSI/IEEE P802.5M-Draft 7, "Source Routing Transparent Bridge
         Operation", IEEE Project 802 (1991).

    [15] ANSI/IEEE 802.1y, "Source Routing Tutorial for End System
         Operation", (September, 1990).

Security Considerations

    Security issues are not discussed in this memo.

Authors' Addresses

    Eric B. Decker
    cisco Systems, Inc.
    1525 O'Brien Dr.
    Menlo Park, CA 94025

    Phone: (415) 326-1941
    Email: cire@cisco.com

    Keith McCloghrie
    Hughes LAN Systems, Inc.
    1225 Charleston Road
    Mountain View, CA 94043

    Phone: (415) 966-7934
    EMail: kzm@hls.com

    Paul Langille
    Digital Equipment Corporation
    Digital Drive, MK02-2/K03
    Merrimack, NH 03054

    Phone: (603) 884-4045
    EMail: langille@edwin.enet.dec.com

    Anil Rijsinghani
    Digital Equipment Corporation
    550 King Street
    Littleton, MA 01460

    Phone: (508) 486-6786
    EMail: anil@levers.enet.dec.com