Network Working Group
Request for Comments: 1243

S. Waldbusser, Editor
Carnegie Mellon University
July 1991

AppleTalk Management Information Base

Status of this Memo

This memo defines objects for managing AppleTalk objects for use with the SNMP protocol. This memo is a product of the AppleTalk-IP Working Group of the Internet Engineering Task Force (IETF). This RFC specifies an IAB standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Abstract

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

2. The Network Management Framework

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

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

RFC 1156 which defines MIB-I, the core set of managed objects for the Internet suite of protocols. RFC 1213, defines MIB-II, an evolution of MIB-I based on implementation experience and new operational requirements.

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.

3. Objects

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) [7] defined in the SMI. In particular, each object has a name, a syntax, and an encoding. The name is an object identifier, an administratively assigned name, which specifies an object type. 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 OBJECT DESCRIPTOR, to also refer to the object type.

The syntax of an object type defines the abstract data structure corresponding to that object type. The ASN.1 language is used for this purpose. However, the SMI [3] purposely restricts the ASN.1 constructs which may be used. These restrictions are explicitly made for simplicity.

The encoding of an object type is simply how that object type is represented using the object type's syntax. Implicitly tied to the notion of an object type's syntax and encoding is how the object type is represented when being transmitted on the network.

The SMI specifies the use of the basic encoding rules of ASN.1 [8], subject to the additional requirements imposed by the SNMP.

3.1. Format of Definitions

Section 5 contains the specification of all object types contained in this MIB module. The object types are defined using the conventions defined in the SMI, as amended by the extensions specified in [9,10].

4. Overview

AppleTalk is a protocol suite which features an open peer-to-peer architecture that runs over a variety of transmission media. AppleTalk is defined in [10]. This protocol suite interoperates with the IP protocol suite through various encapsulation methods. As large AppleTalk networks are built that coexist with large IP networks, a method to manage the AppleTalk networks with SNMP becomes necessary. This MIB defines managed objects to be used for managing AppleTalk networks.

4.1. Structure of MIB

The objects are arranged into the following groups:

- LLAP
- AARP
- ATPort
- DDP
- RTMP
- KIP
- ZIP
- NBP
- ATEcho

These groups are the basic unit of conformance. If the semantics of a group is applicable to an implementation, then it must implement all objects in that group. For example, a managed agent must implement the KIP group if and only if it implements the KIP protocol.

These groups are defined to provide a means of assigning object identifiers, and to provide a method for managed agents to know which objects they must implement.

4.2. The LocalTalk Link Access Protocol Group

The LocalTalk Link Access Protocol (LLAP) is a medium-speed data-link protocol designed for low cost and plug-and-play operation. The LLAP group is designed to manage all interfaces on a managed device that use this protocol.

4.3. The AppleTalk Address Resolution Protocol Group

The AppleTalk Address Resolution Protocol (AARP) is used to map between AppleTalk node addresses, used by the Datagram Delivery Protocol, and the addresses of the underlying data link layer. The AARP table allows for management of the Address Mapping Table on the managed device.

4.4. The AppleTalk Port Group

An AppleTalk Port is a logical connection to a network over which AppleTalk packets can be transmitted. This group allows the management of the configuration of these AppleTalk ports.

4.5. The Datagram Delivery Protocol Group

The Datagram Delivery Protocol (DDP) is the network-layer protocol that is responsible for the socket-to-socket delivery of datagrams over the AppleTalk Internet. This group manages the DDP layer on the managed device.

4.6. The Routing Table Maintenance Protocol Group

The Routing Table Maintenance Protocol (RTMP) is used by AppleTalk routers to create and maintain the routing tables that dictate the process of forwarding datagrams on the AppleTalk internet. The RTMP group manages the RTMP protocol as well as the routing tables generated by this protocol.

4.7. The Kinetics Internet Protocol Group

The Kinetics Internet Protocol (KIP) is a protocol for encapsulating and routing AppleTalk datagrams over an IP internet. This name is historical. The KIP group manages the KIP routing protocol as well as the routing tables generated by this protocol.

4.8. The Zone Information Protocol Group

The Zone Information Protocol (ZIP) is used to maintain a mapping between networks and zone names to facilitate the name lookup process performed by the Name Binding Protocol. The ZIP group manages this protocol and the mapping it produces.

4.9. The Name Binding Protocol Group

The Name Binding Protocol (NBP) is a transport-level protocol that is used to convert human readable service names into the numeric AppleTalk network addresses needed for communicating across the

AppleTalk network. The NBP group manages this protocol and the NBP services that exist on the managed device.

4.10. The AppleTalk Echo Protocol Group

The AppleTalk Echo Protocol is a transport-level protocol used to test and verify the status of the AppleTalk internet. The AtEcho group manages this protocol.

4.11. Textual Conventions

A new data type is introduced as a textual convention in this MIB document. This textual convention enhances the readability of the specification and can ease comparison with other specifications if appropriate. It should be noted that the introduction of this textual convention has no effect on either the syntax or the semantics of any managed objects. The use of this is merely an artifact of the explanatory method used. Objects defined in terms of this method are always encoded by means of the rules that define the primitive type. Hence, no changes to the SMI or the SNMP are necessary to accommodate this textual convention which is adopted merely for the convenience of readers and writers in pursuit of the elusive goal of clear, concise, and unambiguous MIB documents.

The new data type is:

```
DdpAddress ::=          -- 2 octets of net number,
                        -- 1 octet of node number
        OCTET STRING (SIZE (3))
```

5. Definitions

```
RFC1243-MIB DEFINITIONS ::= BEGIN

IMPORTS
        Counter, IpAddress
                FROM RFC1155-SMI
        DisplayString, mib-2
            FROM RFC1213-MIB
        OBJECT-TYPE
                FROM RFC-1212;

--  This MIB module uses the extended OBJECT-TYPE macro as
--  defined in [9]


--  AppleTalk MIB
```

```
            appletalk    OBJECT IDENTIFIER ::= { mib-2 13 }


            DdpAddress ::=  -- 2 octets of net number
                            -- 1 octet of node number
                    OCTET STRING (SIZE (3))
            -- This data type is used for encoding a DDP protocol
            -- address.  The format of this address is a serial
            -- encoding of the two octets of network number in
            -- network byte order, followed by the 1 octet node
            -- number.

            llap     OBJECT IDENTIFIER ::= { appletalk 1 }
            aarp     OBJECT IDENTIFIER ::= { appletalk 2 }
            atport   OBJECT IDENTIFIER ::= { appletalk 3 }
            ddp      OBJECT IDENTIFIER ::= { appletalk 4 }
            rtmp     OBJECT IDENTIFIER ::= { appletalk 5 }
            kip      OBJECT IDENTIFIER ::= { appletalk 6 }
            zip      OBJECT IDENTIFIER ::= { appletalk 7 }
            nbp      OBJECT IDENTIFIER ::= { appletalk 8 }
            atecho   OBJECT IDENTIFIER ::= { appletalk 9 }


        -- The LLAP Group

        llapTable OBJECT-TYPE
                SYNTAX SEQUENCE OF LlapEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "The list of LLAP entries."
                ::= { llap 1 }
        llapEntry OBJECT-TYPE
                SYNTAX LlapEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "An LLAP entry containing objects for the
                    LocalTalk Link Access Protocol for a particular
                    LocalTalk interface."
                INDEX { llapIfIndex }
                ::= { llapTable 1 }

        LlapEntry ::= SEQUENCE {
                llapIfIndex             INTEGER,
                llapInPkts              Counter,
                llapOutPkts             Counter,
                llapInNoHandlers        Counter,
```

```
                llapInLengthErrors      Counter,
                llapInBads              Counter,
                llapCollisions          Counter,
                llapDefers              Counter,
                llapNoDataErrors        Counter,
                llapRandomCTSErrors     Counter,
                llapFCSErrors           Counter
        }
        llapIfIndex OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The LLAP interface to which this entry pertains.
                    The interface identified by a particular value of
                    this index is the same interface as identified
                    by the same value of ifIndex."
                ::= { llapEntry 1 }

        llapInPkts OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of good packets received on this
                    LocalTalk interface."
                ::= { llapEntry 2 }

        llapOutPkts OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of packets transmitted on this
                    LocalTalk interface."
                ::= { llapEntry 3 }

        llapInNoHandlers OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of good packets received on this
                    LocalTalk interface for which there was no
                    protocol handler."
                ::= { llapEntry 4 }
```

```
        llapInLengthErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of packets received on this
                    LocalTalk interface whose actual length did not
                    match the length in the header."
                ::= { llapEntry 5 }

        llapInErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of packets containing errors
                    received on this LocalTalk interface."
                ::= { llapEntry 6 }

        llapCollisions OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of collisions assumed on this
                    LocalTalk interface due to the lack of a lapCTS
                    reply."
                ::= { llapEntry 7 }

        llapDefers OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of times this LocalTalk
                    interface deferred to other packets."
                ::= { llapEntry 8 }

        llapNoDataErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of times this LocalTalk
                    interface received a lapRTS packet and expected
                    a data packet, but did not receive any data
                    packet."
                ::= { llapEntry 9 }
```

```
        llapRandomCTSErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of times this LocalTalk
                    interface received a lapCTS packet that was
                    not solicited by a lapRTS packet."
                ::= { llapEntry 10 }

        llapFCSErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of times this LocalTalk
                    interface received a packet with an FCS
                    (Frame Check Sequence) error."
                ::= { llapEntry 11 }


        -- The AARP Group
        aarpTable OBJECT-TYPE
                SYNTAX SEQUENCE OF AarpEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "The AppleTalk Address Translation Table
                    contains an equivalence of AppleTalk Network
                    Addresses to the link layer physical address."
                ::= { aarp 1 }

        aarpEntry OBJECT-TYPE
                SYNTAX AarpEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "Each entry contains one AppleTalk Network
                    Address to physical address equivalence."
                INDEX { aarpIfIndex, aarpNetAddress }
                ::= { aarpTable 1 }

        AarpEntry ::= SEQUENCE {
                aarpIfIndex     INTEGER,
                aarpPhysAddress OCTET STRING,
                aarpNetAddress  DdpAddress
        }
```

```
        aarpIfIndex OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The interface on which this entry's equivalence
                    is effective.  The interface identified by a
                    particular value of this index is the same
                    interface as identified by the same value of
                    ifIndex."
                ::= { aarpEntry 1 }

        aarpPhysAddress OBJECT-TYPE
                SYNTAX OCTET STRING
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The media-dependent physical address"
                ::= { aarpEntry 2 }

        aarpNetAddress OBJECT-TYPE
                SYNTAX DdpAddress
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The AppleTalk Network Address corresponding to
                    the media-dependent physical address."
                ::= { aarpEntry 3 }


        -- The ATPort Group

        atportTable OBJECT-TYPE
                SYNTAX SEQUENCE OF AtportEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "A list of AppleTalk ports for this entity."
                ::= { atport 1 }

        atportEntry OBJECT-TYPE
                SYNTAX AtportEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "The description of one of the AppleTalk
                    ports on this entity."
                INDEX { atportIndex }
```

```
            ::= { atportTable 1 }

        AtportEntry ::= SEQUENCE {
                atportIndex                INTEGER,
                atportDescr                DisplayString,
                atportType                 INTEGER,
                atportNetStart             OCTET STRING (SIZE(2)),
                atportNetEnd               OCTET STRING (SIZE(2)),
                atportNetAddress           DdpAddress,
                atportStatus               INTEGER,
                atportNetConfig            INTEGER,
                atportZoneConfig           INTEGER,
                atportZone                 OCTET STRING,
                atportIfIndex              INTEGER
        }

        atportIndex OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "A unique value for each AppleTalk port.
                    Its value is between 1 and the total number of
                    AppleTalk ports.  The value for each port must
                    remain constant at least from the
                    re-initialization of the entity's network
                    management system to the next
                    re-initialization."
                ::= { atportEntry 1 }

        atportDescr OBJECT-TYPE
                SYNTAX DisplayString
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "A text string containing information about the
                    port.  This string is intended for presentation
                    to a human; it must not contain anything but
                    printable ASCII characters."
                ::= { atportEntry 2 }

        atportType OBJECT-TYPE
                SYNTAX INTEGER {
                     other(1),       -- none of the following
                     localtalk(2),
                     ethertalk1(3),
                     ethertalk2(4),
                     tokentalk(5),
```

```
                    iptalk(6),
                    serial-ppp(7),
                    serial-nonstandard(8),
                    virtual(9)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The type of port, distinguished by the protocol
                    immediately below DDP in the protocol stack."
                ::= { atportEntry 3 }

        atportNetStart OBJECT-TYPE
                SYNTAX OCTET STRING (SIZE(2))
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The first AppleTalk network address in the range
                    configured for this port.  This is a two octet
                    DDP network address in network byte order."
                ::= { atportEntry 4 }

        atportNetEnd OBJECT-TYPE
                SYNTAX OCTET STRING (SIZE(2))
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The last AppleTalk network address in the range
                    configured for this port.  This is a two octet
                    DDP network address in network byte order.  If the
                    network to which this AppleTalk port is
                    connected is a Phase 1 network or a non-extended
                    network, the value for atportNetEnd shall be two
                    octets of zero."
                ::= { atportEntry 5 }

        atportNetAddress OBJECT-TYPE
                SYNTAX DdpAddress
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The AppleTalk network address configured for this
                    port."
                ::= { atportEntry 6 }

        atportStatus OBJECT-TYPE
                SYNTAX INTEGER {
                    operational(1),
```

```
                    unconfigured(2),
                    off(3),
                    invalid(4)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The configuration status of this port.

                    Setting this object to the value invalid(4)
                    has the effect of invalidating the corresponding
                    entry in the atportTable.  That is, it
                    effectively disassociates the mapping identified
                    with said entry.  It is an
                    implementation-specific matter as to whether the
                    agent removes an invalidated entry from the table.
                    Accordingly, management stations must be
                    prepared to receive from agents tabular
                    information corresponding to entries not
                    currently in use.  Proper interpretation of such
                    entries requires examination of the relevant
                    atportStatus object."
                        ::= { atportEntry 7 }

        atportNetConfig OBJECT-TYPE
                SYNTAX INTEGER {
                    configured(1),  -- explicit configuration.
                    garnered(2),    -- assumed from inspection of net.
                    guessed(3),     -- a "random" configuration.
                    unconfigured(4)
                }
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The configuration status of this port."
                ::= { atportEntry 8 }

        atportZoneConfig OBJECT-TYPE
                SYNTAX INTEGER {
                    configured(1),  -- explicit configuration
                    garnered(2),    -- assumed from inspection of net.
                    guessed(3),     -- a "random" configuration.
                    unconfigured(4)
                }
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The configuration status of the zone information
```

```
                for this port."
            ::= { atportEntry 9 }

        atportZone OBJECT-TYPE
                SYNTAX OCTET STRING
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The zone name configured for this AppleTalk
                    port."
                ::= { atportEntry 10 }

        atportIfIndex OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The physical interface associated with this
                    AppleTalk port.  The interface identified by a
                    particular value of this index is the same
                    interface as identified by the same value of
                    ifIndex."
                ::= { atportEntry 11 }


        -- The DDP Group

        ddpOutRequests OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of DDP datagrams which were
                    supplied to DDP by local DDP clients in requests
                    for transmission.  Note that this counter does
                    not include any datagrams counted in
                    ddpForwRequests."
                ::= { ddp 1 }

        ddpOutShorts OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of short DDP datagrams which
                    were transmitted from this entity."
                ::= { ddp 2 }
```

```
        ddpOutLongs OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of long DDP datagrams which were
                    transmitted from this entity."
                ::= { ddp 3 }

        ddpInReceives OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of input datagrams received by
                    DDP, including those received in error."
                ::= { ddp 4 }

        ddpForwRequests OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The number of input datagrams for which this
                    entity was not their final DDP destination, as
                    a result of which an attempt was made to find a
                    route to forward them to that final destination."
                ::= { ddp 5 }

        ddpInLocalDatagrams OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of input DDP datagrams for
                    which this entity was their final DDP
                    destination."
                ::= { ddp 6 }

        ddpNoProtocolHandlers OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of DDP datagrams addressed to
                    this entity that were addressed to an upper
                    layer protocol for which no protocol handler
                    existed."
```

```
            ::= { ddp 7 }

        ddpOutNoRoutes OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                "The total number of DDP datagrams dropped
                because a route could not be found to their
                final destination."
            ::= { ddp 8 }

        ddpTooShortErrors OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                "The total number of input DDP datagrams dropped
                because the received data length was less than
                the data length specified in the DDP header or
                the received data length was less than the
                length of the expected DDP header."
            ::= { ddp 9 }

        ddpTooLongErrors OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                "The total number of input DDP datagrams dropped
                because the received data length was greater
                than the data length specified in the DDP header
                or because they exceeded the maximum DDP
                datagram size."
            ::= { ddp 10 }

        ddpBroadcastErrors OBJECT-TYPE
            SYNTAX Counter
            ACCESS read-only
            STATUS mandatory
            DESCRIPTION
                "The total number of input DDP datagrams dropped
                because this entity was not their final
                destination and they were addressed to the link
                level broadcast."
            ::= { ddp 11 }
```

```
        ddpShortDDPErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of input DDP datagrams dropped
                    because this entity was not their final
                    destination and their type was short DDP."
                ::= { ddp 12 }

        ddpHopCountErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of input DDP datagrams dropped
                    because this entity was not their final
                    destination and their hop count would exceed 15."
                ::= { ddp 13 }

        ddpChecksumErrors OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The total number of input DDP datagrams dropped
                    because of a checksum error."
                ::= { ddp 14 }


        -- The RTMP Group

        rtmpTable OBJECT-TYPE
                SYNTAX SEQUENCE OF RtmpEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "A list of Routing Table Maintenance Protocol
                    entries for this entity."
                ::= { rtmp 1 }

        rtmpEntry OBJECT-TYPE
                SYNTAX RtmpEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "The route entry to a particular network range."
                INDEX { rtmpRangeStart }
```

```
            ::= { rtmpTable 1 }

        RtmpEntry ::= SEQUENCE {
                rtmpRangeStart  OCTET STRING (SIZE(2)),
                rtmpRangeEnd    OCTET STRING (SIZE(2)),
                rtmpNextHop     OCTET STRING,
                rtmpType        INTEGER,
                rtmpPort        INTEGER,
                rtmpHops        INTEGER,
                rtmpState       INTEGER
        }

        rtmpRangeStart OBJECT-TYPE
                SYNTAX OCTET STRING (SIZE(2))
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The first DDP network address in the network
                    range to which this routing entry pertains.
                    This is a two octet DDP network address in
                    network byte order."
                ::= { rtmpEntry 1 }

        rtmpRangeEnd OBJECT-TYPE
                SYNTAX OCTET STRING (SIZE(2))
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The last DDP network address in the network range
                    to which this routing entry pertains.  This is a
                    two octet DDP network address in network byte
                    order.  If the network to which this routing
                    entry pertains is a Phase 1 network or a
                    non-extended network, the value for rtmpRangeEnd
                    shall be two octets of zero."
                ::= { rtmpEntry 2 }

        rtmpNextHop OBJECT-TYPE
                SYNTAX OCTET STRING
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The next hop in the route to this entry's
                    destination network.  If the type of this route
                    is Appletalk, this address takes the same form
                    as DdpAddress."
                ::= { rtmpEntry 3 }
```

```
        rtmpType OBJECT-TYPE
                SYNTAX INTEGER {
                        other(1),
                        appletalk(2),
                        serial-ppp(3),
                        serial-nonstandard(4)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The type of network over which this route
                    points."
                ::= { rtmpEntry 4 }

        rtmpPort OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The index of the AppleTalk port over which
                    this route points."
                ::= { rtmpEntry 5 }

        rtmpHops OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The number of hops required to reach the
                    destination network to which this routing
                    entry pertains."
                ::= { rtmpEntry 6 }

        rtmpState OBJECT-TYPE
                SYNTAX INTEGER {
                    good(1),
                    suspect(2),
                    goingBad(3),
                    bad(4) -- may be removed from table
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The status of the information contained in this
                    route entry.

                    Setting this object to the value bad(4) has the
                    effect of invalidating the corresponding entry
```

```
                in the rtmpTable.  That is, it  effectively
                disassociates the mapping identified with said
                entry.  It is an implementation-specific matter
                as to whether the agent removes an invalidated
                entry from the table.  Accordingly, management
                stations must be prepared to receive from agents
                tabular information corresponding to entries not
                currently in use.  Proper interpretation of such
                entries requires examination of the relevant
                rtmpState object."
            ::= { rtmpEntry 7 }


        -- The KIP Group

        kipTable OBJECT-TYPE
                SYNTAX SEQUENCE OF KipEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "The table of routing information for KIP
                    networks."
                ::= { kip 1 }

        kipEntry OBJECT-TYPE
                SYNTAX KipEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "An entry in the routing table for KIP networks."
                INDEX { kipNetStart }
                ::= { kipTable 1 }

        KipEntry ::= SEQUENCE {
                kipNetStart     OCTET STRING (SIZE(2)),
                kipNetEnd       OCTET STRING (SIZE(2)),
                kipNextHop      IpAddress,
                kipHopCount     INTEGER,
                kipBCastAddr    IpAddress,
                kipCore         INTEGER,
                kipType         INTEGER,
                kipState        INTEGER,
                kipShare        INTEGER
        }

        kipNetStart OBJECT-TYPE
                SYNTAX OCTET STRING (SIZE(2))
                ACCESS read-write
```

```
                STATUS mandatory
                DESCRIPTION
                    "The first AppleTalk network address in the
                    range for this routing entry.  This address is a
                    two octet DDP network address in network byte
                    order."
                ::= { kipEntry 1 }

        kipNetEnd OBJECT-TYPE
                SYNTAX OCTET STRING (SIZE(2))
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The last AppleTalk network address in the range
                    for this routing entry.  This address is a two
                    octet DDP network address in network byte order.
                    If the network to which this AppleTalk port is
                    connected is a Phase 1 network or a non-extended
                    network, the value for kipNetEnd shall be two
                    octets of zero."
                ::= { kipEntry 2 }

        kipNextHop OBJECT-TYPE
                SYNTAX IpAddress
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The IP address of the next hop in the route to
                    this entry's destination network."
                ::= { kipEntry 3 }

        kipHopCount OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The number of hops required to reach the
                    destination network to which this entry pertains."
                ::= { kipEntry 4 }

        kipBCastAddr OBJECT-TYPE
                SYNTAX IpAddress
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The form of the IP address used to broadcast on
                    this network."
                ::= { kipEntry 5 }
```

```
        kipCore OBJECT-TYPE
                SYNTAX INTEGER {
                     core(1),
                     notcore(2)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The status of this network as a Kip Core
                    network."
                ::= { kipEntry 6 }

        kipType OBJECT-TYPE
                SYNTAX INTEGER {
                     kipRouter(1),
                     net(2),
                     host(3),
                     other(4)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The type of the entity that this route points
                    to."
                ::= { kipEntry 7 }

        kipState OBJECT-TYPE
                SYNTAX INTEGER {
                     configured(1),
                     learned(2),
                     invalid(3)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The state of this network entry.

                    Setting this object to the value invalid(3) has
                    the effect of invalidating the corresponding
                    entry in the kipTable.  That is, it effectively
                    disassociates the mapping identified with said
                    entry.  It is an implementation-specific matter
                    as to whether the agent removes an invalidated
                    entry from the table.
                    Accordingly, management stations must be
                    prepared to receive from agents tabular
                    information corresponding to entries not
                    currently in use.  Proper interpretation of such
```

```
                entries requires examination of the relevant
                kipState object."
            ::= { kipEntry 8 }

        kipShare OBJECT-TYPE
                SYNTAX INTEGER {
                     shared(1),
                     private(2)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "If the information in this entry is propagated
                    to other routers as part of a routing protocol,
                    the value of this variable is equal to
                    shared(1).  Otherwise its value is private(2)."
                ::= { kipEntry 9 }


        -- The ZIP Group

        zipTable OBJECT-TYPE
                SYNTAX SEQUENCE OF ZipEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "The table of zone information for reachable
                    AppleTalk networks."
                ::= { zip 1 }

        zipEntry OBJECT-TYPE
                SYNTAX ZipEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "An entry of zone information for a particular
                    zone and network combination."
                INDEX { zipZoneNetStart, zipZoneIndex }
                ::= { zipTable 1 }

        ZipEntry ::= SEQUENCE {
                zipZoneName     OCTET STRING,
                zipZoneIndex    INTEGER,
                zipZoneNetStart OCTET STRING (SIZE(2)),
                zipZoneNetEnd   OCTET STRING (SIZE(2)),
                zipZoneState    INTEGER
        }
```

```
        zipZoneName OBJECT-TYPE
                SYNTAX OCTET STRING
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The ASCII zone name of this entry."
                ::= { zipEntry 1 }

        zipZoneIndex OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "An integer that is unique to the zipZoneName
                    that is present in this entry.  For any given
                    zone name, every zipEntry that has an equal zone
                    name will have the same zipZoneIndex."
                ::= { zipEntry 2 }

        zipZoneNetStart OBJECT-TYPE
                SYNTAX OCTET STRING (SIZE(2))
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The network that starts the range for this
                    entry.  This address is a two octet DDP network
                    address in network byte order."
                ::= { zipEntry 3 }

        zipZoneNetEnd OBJECT-TYPE
                SYNTAX OCTET STRING (SIZE(2))
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The network that ends the range for this
                    entry.  This address is a two octet DDP network
                    address in network byte order.  If the network
                    to which this zip entry pertains is a Phase 1
                    network or a non-extended network, the value for
                    zipZoneNetEnd shall be two bytes of zero."
                ::= { zipEntry 4 }

        zipZoneState OBJECT-TYPE
                SYNTAX INTEGER {
                        valid(1),
                        invalid(2)
                }
                ACCESS read-write
```

```
                STATUS mandatory
                DESCRIPTION
                    "The state of this zip entry.

                    Setting this object to the value invalid(2) has
                    the effect of invalidating the corresponding
                    entry in the zipTable.  That is, it effectively
                    disassociates the mapping identified with said
                    entry.  It is an implementation-specific matter
                    as to whether the agent removes an invalidated
                    entry from the table.
                    Accordingly, management stations must be
                    prepared to receive from agents tabular
                    information corresponding to entries not
                    currently in use.  Proper interpretation of
                    such entries requires examination of the
                    relevant zipZoneState object."
                ::= { zipEntry 5 }


        -- The NBP Group

        nbpTable OBJECT-TYPE
                SYNTAX SEQUENCE OF NbpEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "The table of NBP services registered on this
                    entity."
                ::= { nbp 1 }

        nbpEntry OBJECT-TYPE
                SYNTAX NbpEntry
                ACCESS not-accessible
                STATUS mandatory
                DESCRIPTION
                    "The description of an NBP service registered on
                    this entity."
                INDEX { nbpIndex }
                ::= { nbpTable  1 }

         NbpEntry ::= SEQUENCE {
                nbpIndex        INTEGER,
                nbpObject       OCTET STRING,
                nbpType         OCTET STRING,
                nbpZone         OCTET STRING,
                nbpState        INTEGER
         }
```

```
        nbpIndex OBJECT-TYPE
                SYNTAX INTEGER
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The index of this NBP entry.  This value ranges
                    from 1 to the number of NBP entries currently
                    registered on this entity."
                ::= { nbpEntry 1 }

        nbpObject OBJECT-TYPE
                SYNTAX OCTET STRING
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The name of the service described by this
                    entity."
                ::= { nbpEntry 2 }

        nbpType OBJECT-TYPE
                SYNTAX OCTET STRING
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The type of the service described by this
                    entity."
                ::= { nbpEntry 3 }

        nbpZone OBJECT-TYPE
                SYNTAX OCTET STRING
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The zone the service described by this entity is
                    registered in."
                ::= { nbpEntry 4 }

        nbpState OBJECT-TYPE
                SYNTAX INTEGER {
                        valid(1),
                        invalid(2)
                }
                ACCESS read-write
                STATUS mandatory
                DESCRIPTION
                    "The state of this NBP entry.

                    Setting this object to the value invalid(2) has
```

```
                    the effect of invalidating the corresponding
                    entry in the nbpTable.  That is, it effectively
                    disassociates the mapping identified with said
                    entry.  It is an implementation-specific matter
                    as to whether the agent removes an invalidated
                    entry from the table.
                    Accordingly, management stations must be
                    prepared to receive from agents tabular
                    information corresponding to entries not
                    currently in use.  Proper interpretation of
                    such entries requires examination of the
                    relevant nbpState object."
                ::= { nbpEntry 5 }


        -- The ATEcho Group

        atechoRequests OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The number of AppleTalk echo requests received."
                ::= { atecho 1 }

        atechoReplies OBJECT-TYPE
                SYNTAX Counter
                ACCESS read-only
                STATUS mandatory
                DESCRIPTION
                    "The number of AppleTalk echo replies sent."
                ::= { atecho 2 }
        END
```

6. Acknowledgements

This document was produced by the IETF AppleTalk-IP Working Group:

Terry Braun, Novell
Gregory Bruell, Shiva
Philip Budne, Shiva
Rob Chandhok, CMU
Cyrus Chow, NASA
Bruce Crabill, UMD
Peter DiCamillo, Brown
Robert Elz, U. of Melbourne
Tom Evans, Webster
Karen Frisa, CMU

Russ Hobby, UC Davis
Tom Holodnik, CMU
Peter Honeyman, U. of Michigan
Michael Horowitz, Shiva
Van Jacobson, Lawrence Berkeley Labs
Doug Kerr, Novell
Holly Knight, Apple
Philip Koch, Dartmouth
Louise Laier, Apple
Nik Langrind, Shiva
Joshua Littlefield, Cayman
Kanchei Loa, Motorola
John Mason, Apple
Leo McLaughlin, TWG
Milo Medin, NASA
Greg Minshall, Novell
Bob Morgan, Stanford
Ed Moy, Berkeley
Matthew Nocifore, Drexel
Zbigniew Opalka, BBN
Alan Oppenheimer, Apple
Brad Parker, Cayman
Greg Satz, Cisco
John Seligson, Apple
Frank Slaughter, Shiva
Zaw-Sing Su, SRZ
John Veizades, Apple
Peter Vinsel, Apple
Jonathan Wenocur, Shiva
Steven Willis, Wellfleet

In addition, the contribution of the following individuals is also acknowledged:

Karen Frisa, Carnegie Mellon University
Greg Minshall, Novell, Inc.
Marshall T. Rose, PSI

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", RFC 1155,

Performance Systems International, Hughes LAN Systems, May 1990.

[4] McCloghrie K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1156, Hughes LAN Systems, Performance Systems International, May 1990.

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

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

[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", RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.

[10] Sidhu, G., Andrews, R., and A. Oppenheimer, "Inside AppleTalk", Second Edition, Addison Wesley, 1990.

8. Security Considerations

Security issues are not discussed in this memo.

9. Author's Address

Steven Waldbusser
Carnegie Mellon University
4910 Forbes Ave.
Pittsburgh, PA 15213

EMail: waldbusser@andrew.cmu.edu