Internet Engineering Task Force
INTERNET-DRAFT

Nevil Brownlee
The University of Auckland
Feb 1996

Traffic Flow Measurement: Meter MIB

<draft-ietf-rtfm-acct-meter-mib-00.txt>

Status of this Memo

Abstract

This memo defines an experimental portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP-based internets. In particular, this memo defines managed objects used for obtaining traffic flow information from network traffic meters.

Contents

1 The Network Management Framework

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

RFC 1155 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 defines MIB-I, the core set of managed objects for the Internet suite of protocols. RFC 1213 [1] defines MIB-II, an evolution of MIB-I based on implementation experience and new operational requirements.

RFC 1157 defines the SNMP, the protocol used for network access to managed objects.

RFC 1442 [2] defines the SMI for version 2 of the Simple Network Management Protocol.

RFCs 1443 and 1444 [3,4] define Textual Conventions and Conformance Statements for version 2 of the Simple Network Management Protocol.

RFC 1452 [5] describes how versions 1 and 2 of the Simple Network Management Protocol should coexist.

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

2 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) [6] 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 [2] 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 [7], subject to the additional requirements imposed by the SNMP.

2.1 Format of Definitions

Section 4 contains contains the specification of all object types contained in this MIB module. These object types are defined using the conventions defined in [2] and [3].

3 Overview

Traffic Flow Measurement seeks to provide a well-defined method for gathering traffic flow information from networks and internetworks. The background for this is given in "Traffic Flow Measurement: Background" [8]. The Realtime Traffic Flow Measurement (rtfm) Working Group has produced a measurement architecture to achieve it; this is documented in "Traffic Flow Measurement: Architecture" [9]. The architecture defines three entities:

- METERS, which observe network traffic flows and build up a table of flow data records for them,

- METER REAERS, which collect traffic flow data from meters, and

- MANAGERS, which oversee the operation of meters and meter readers.

This memo defines the SNMP management information for a Traffic Flow Meter (TFM). It documents the earlier work of the Internet Accounting Working Group, and is intended to provide a starting point for the Realtime Traffic Flow Measurement Working Group.

3.1 Scope of Definitions, Textual Conventions

All objects defined in this memo are registered in a single subtree within the mib-2 namespace [1,2], and are for use in network devices which may perform a PDU forwarding or monitoring function. For these devices, the value of the ifSpecific variable in the MIB-II [1] has the OBJECT IDENTIFIER value:

flowMIB OBJECT IDENTIFIER ::= mib-2 40

as defined below.

The RTFM Meter MIB was first produced and tested using SNMPv1. It has been converted into SNMPv2 following the guidelines in RFC 1452 [5].

3.2 Usage of the MIB variables

The MIB breaks into four parts - control, flows, rules and conformance statements.

The rules implement the minumum set of packet-matching actions, as set out in the "Traffic Flow Measurment: Architecture" document [9]. In addition they provide for BASIC-style subroutines, allowing a network manager to dramatically reduce the number of rules required to monitor a big network.

Traffic flows are identified by a set of attributes for each of its end-points. Attributes include network addresses for each layer of the network protocol stack, and 'subscriber ids,' which may be used to identify an accountable entity for the flow.

The conformance statements are set out as defined in [4]. They explain

what must be implemented in a meter which claims to conform to this MIB.

To retrieve flow data one could simply do a linear scan of the flow table. This would certainly work, but would require a lot of protocol exchanges. To reduce the overhead in retrieving flow data, there are two 'indexes' into the flow table. The 'activity' index makes it easy to find those flows which have been active at or after a given time; this allows retrieval of flow data without using an opaque object.

The other index is the flowColumnActivityTable, which is (logically) a three-dimensional array, subscripted by flow attribute, activity time and starting flow number. This allows a meter reader to retrieve (in an opaque object) data for a column of the flow table with a minimum of SNMP overhead. An attempt has been made to include a full ASN.1 definition of the flowColumnActivityData object.

One aspect of data collection which needs emphasis is that all the MIB variables are set up to allow multiple independent colletors to work properly, i.e. the flow table indexes are stateless. An alternative approach would have been to 'snapshot' the flow table, which would mean that the meter readers would have to be synchronized. The stateless approach does mean that two meter readers will never return exactly the same set of traffic counts, but over long periods (e.g. 15-minute collections over a day) the discrepancies are acceptable. If one really needs a snapshot, this can be achieved by switching to an identical rule set with a different RuleSet number, hence asynchronous collections may be regarded as a useful generalisation of synchronised ones.

The control variables are the minimum set required for a meter reader. Their number has been whittled down as experience has been gained with the MIB implementation.

4 Definitions

```
FLOW-METER-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Counter32, Integer32, TimeTicks,
    IpAddress
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, RowStatus, TimeStamp
        FROM SNMPv2-TC
    OBJECT-GROUP, MODULE-COMPLIANCE
        FROM SNMPv2-CONF
    mib-2, ifIndex
        FROM RFC1213-MIB;

flowMIB MODULE-IDENTITY
    LAST-UPDATED "9602080845Z"
```

```
    ORGANIZATION "IETF Realtime Traffic Flow Measurement Working Group"
```

```
    CONTACT-INFO
        "Nevil Brownlee, The University of Auckland
        Email: n.brownlee@auckland.ac.nz"
    DESCRIPTION
                "MIB for the RTFM Traffic Flow Meter."
    ::= { mib-2 40 }


flowControl         OBJECT IDENTIFIER ::= { flowMIB 1 }

flowData            OBJECT IDENTIFIER ::= { flowMIB 2 }

flowRules           OBJECT IDENTIFIER ::= { flowMIB 3 }

flowMIBConformance  OBJECT IDENTIFIER ::= { flowMIB 4 }


-- Textual Conventions

AddressType ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
        "Indicates the type of an adjacent address or peer address.
        The type of a transport address will depend on the peer
        address type."
    SYNTAX  INTEGER {
        ethernetAddress(1),
        ipAddress(2),
        nsapAddress(3),
        idprAddress(4),
        decnetAddress(5),
        ipxnetAddress(6),
        ethertalkAddress(7),
        fddiAddress(8),
        tokenringAddress(9) }

AdjacentAddress ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
        "Specifies the value of an adjacent address for various
        physical media.  Address format depends on the actual media,
        as follows:

        Ethernet:    ethernetAddress(1)
            6-octet 802.3 MAC address in 'canonical' order

        FDDI:        fddiAddress(3)
            FddiMACLongAddress, i.e. a 6-octet MAC address
            in 'canonical' order  (defined in the FDDI MIB [10])
```

Token Ring: tokenringAddress(4)

```
            6-octet 802.5 MAC address in 'canonical' order
        "
    SYNTAX OCTET STRING (SIZE (6))

PeerAddress ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
        "Specifies the value of a peer address for various network
        protocols.  Address format depends on the actual protocol,
        as follows:

        IP:          ipAddress(2)
            4-octet IpAddress  (defined in the SNMPv2 SMI [2])

        CLNS:        nsapAddress(3)
            NsapAddress  (defined in the SNMPv2 SMI [2])

        IDRP:        idprAddress(4)
            InterDomain Routing Protocol (a version of BGP)

        DECnet:      decnetAddress(5)
            1-octet Area number (in low-order six bits),
            2-octet Host number (in low-order ten bits)

        Novell:      ipxnetAddress(6)
            4-octet Network number,
            6-octet Host number (MAC address)

        AppleTalk:   ethertalkAddress(7)
            2-octet Network number (sixteen bits),
            1-octet Host number (eight bits)
        "
    SYNTAX OCTET STRING (SIZE (3..20))

TransportAddress ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
        "Specifies the value of a transport address for various
        network protocols.  Format as follows:

        IP:
            2-octet UDP or TCP port number

        Other protocols:
            2-octet port number
        "
    SYNTAX OCTET STRING (SIZE (2))

RuleAddress ::= TEXTUAL-CONVENTION
    STATUS  current
```

DESCRIPTION

```
        "Specifies the value of an address.  Is a superset of
        AdjacentAddress, PeerAddress and TransportAddress."
    SYNTAX OCTET STRING (SIZE (2..20))

FlowAttributeNumber ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
        "Uniquely identifies an attribute within a flow data record."
    SYNTAX  INTEGER {
        flowIndex(1),
        flowStatus(2),
        sourceInterface(3),       -- Source Address
        sourceAdjacentType(4),
        sourceAdjacentAddress(5),
        sourceAdjacentMask(6),
        sourcePeerType(7),
        sourcePeerAddress(8),
        sourcePeerMask(9),
        sourceTransType(10),
        sourceTransAddress(11),
        sourceTransMask(12),
        destInterface(13),        -- Dest Address
        destAdjacentType(14),
        destAdjacentAddress(15),
        destAdjacentMask(16),
        destPeerType(17),
        destPeerAddress(18),
        destPeerMask(19),
        destTransType(20),
        destTransAddress(21),
        destTransMask(22),
        pduScale(23),             -- Rule Set attributes
        octetScale(24),
        ruleSet(25),
        toOctets(26),             -- Source-to-Dest
        toPDUs(27),
        fromOctets(28),           -- Dest-to-Source
        fromPDUs(29),
        firstTime(30),            -- Activity times
        lastActiveTime(31),
        sourceSubscriberID(32),   -- Subscriber ID
        destSubscriberID(33),
        sessionID(34),
        sourceClass(35),          -- Computed attributes
        destClass(36),
        flowClass(37),
        sourceKind(38),
        destKind(39),
        flowKind(40) }
```

```
RuleAttributeNumber ::= TEXTUAL-CONVENTION
```

```
    STATUS  current
    DESCRIPTION
        "Uniquely identifies an attribute which may be tested in
        a rule.  These include attributes whose values come directly
        from the flow's packets and the five 'meter' variables used to
        hold an AttributeValue.  Attributes derived from the rules -
        e.g. address masks - may not be tested."
    SYNTAX  INTEGER {
        null(0),
        sourceInterface(3),       -- Source Address
        sourceAdjacentType(4),
        sourceAdjacentAddress(5),
        sourcePeerType(7),
        sourcePeerAddress(8),
        sourceTransType(10),
        sourceTransAddress(11),
        destInterface(13),        -- Dest Address
        destAdjacentType(14),
        destAdjacentAddress(15),
        destPeerType(17),
        destPeerAddress(18),
        destTransType(20),
        destTransAddress(21),
        sourceSubscriberID(32),   -- Subscriber ID
        destSubscriberID(33),
        sessionID(34),
        v1(51),                   -- Meter variables
        v2(52),
        v3(53),
        v4(54),
        v5(55) }

TimeFilter ::= TEXTUAL-CONVENTION
    STATUS  current
    DESCRIPTION
        "Used as an index to a table.  A TimeFilter variable allows
        a GetNext or GetBulk request to find rows in a table for
        which the TimeFilter index variable is greater than or equal
        to a specified value.  For example, a meter reader could
        use the FlowActivityTable to find the subscripts for all
        rows in the flow table which have been active at or since
        a specified time.

        More details on TimeFilter variables, their implementation
        and use can be found in the RMON2 MIB [11]."
    SYNTAX  TimeTicks

ActionNumber ::= TEXTUAL-CONVENTION
    STATUS  current
```

```
    DESCRIPTION
        "Uniquely identifies the action of a rule, i.e. the Pattern
```

```
        Matching Engine's opcode number.  Details of the opcodess
        are given in the 'Traffic Flow Measurement: Architecture'
        document [9]."
    SYNTAX  INTEGER {
        ignore(1),
        fail(2),
        count(3),
        countPkt(4),
        return(5),
        gosub(6),
        gosubAct(7),
        assign(8),
        assignAct(9),
        goto(10),
        gotoAct(11),
        pushRuleTo(12),
        pushRuleToAct(13),
        pushPktTo(14),
        pushPktToAct(15) }

--
-- Control Group:  Rule Set Info Table
--

flowRuleSetInfoTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FlowRuleSetInfoEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "An array of information about the rule sets held in the
        meter.   Rule set 1 is the meter default, used when the meter
        starts up.  It is built in to the meter; it may not be
        changed."
    ::= { flowControl 1 }

flowRuleSetInfoEntry OBJECT-TYPE
    SYNTAX  FlowRuleSetInfoEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "Information about a particular rule set."
    INDEX  { flowRuleInfoIndex }
    ::= { flowRuleSetInfoTable 1 }

FlowRuleSetInfoEntry ::= SEQUENCE {
    flowRuleInfoIndex         Integer32,
    flowRuleInfoSize          Integer32,
    flowRuleInfoOwner         IpAddress,
    flowRuleInfoTimeStamp     TimeStamp,
```

```
    flowRuleInfoStatus        RowStatus
    }
```

```
flowRuleInfoIndex OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "An index which selects an entry in the flowRuleSetInfoTable.
        Each such entry contains control information for a particular
        rule set which the meter may run."
    ::= { flowRuleSetInfoEntry 1 }

flowRuleInfoSize OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "Number of rules in this rule set.  Setting this variable will
        cause the meter to allocate space for these rules."
    ::= { flowRuleSetInfoEntry 2 }

flowRuleInfoOwner OBJECT-TYPE
    SYNTAX  IpAddress
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "Identifies the meter reader which configured this rule set."
    ::= { flowRuleSetInfoEntry 3 }

flowRuleInfoTimeStamp OBJECT-TYPE
    SYNTAX  TimeStamp
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "Time this rule set was last changed."
    ::= { flowRuleSetInfoEntry 4 }

flowRuleInfoStatus OBJECT-TYPE
    SYNTAX  RowStatus
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "The status of this rule set.  If this object's value is
        not active(1), the meter will not attempt to use this
        rule set."
    ::= { flowRuleSetInfoEntry 5 }

--
-- Control Group:  Interface Info Table
--
```

```
flowInterfaceTable OBJECT-TYPE
```

```
    SYNTAX  SEQUENCE OF FlowInterfaceEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "An array of information specific to each meter interface."
    ::= { flowControl 2 }

flowInterfaceEntry OBJECT-TYPE
    SYNTAX  FlowInterfaceEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "Information about a particular interface."
    INDEX   { ifIndex }
    ::= { flowInterfaceTable 1 }

FlowInterfaceEntry ::= SEQUENCE {
    flowInterfaceRate         Integer32,
    flowInterfaceLostPackets  Counter32
    }

flowInterfaceRate OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "The parameter N for statistical counting on this interface.
        Set to N to count 1/Nth of the packets appearing at this
        interface.  A meter should choose its own algorithm to
        introduce variance into the sampling so that exactly every Nth
        packet is not counted.  A sampling rate of 1 yields a normal
        counter.  A sampling rate of 0 results in the interface
        being ignored by the meter."
    DEFVAL { 1 } -- Count every packet,
    ::= { flowInterfaceEntry 1 }

flowInterfaceLostPackets OBJECT-TYPE
    SYNTAX  Counter32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The number of packets the meter has lost for this interface.
        Such losses may occur because the meter's network interface
        hardware or software has been unable to keep up with the
        traffic volume or because the meter has run out of memory
        for new flows."
    ::= { flowInterfaceEntry 2 }

--
```

```
-- Control Group:  Meter Reader Info Table
--
```

```
-- At present any meter reader wishing to collect flow data may do so
-- by writing the flowLastReadTime object.   The meter interprets
-- such a write as the start of a new collection and updates the meter
-- reader's row in the reader info table (creating a new row if
-- neccessary).  In future it may be better to have meter readers
-- explicitly create a row in the reader info table, and indicate
-- that they are starting a collection by writing that row's
-- flowReaderLastTime object."

flowReaderInfoTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FlowReaderInfoEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "An array of information about meter readers which may
        collect flow data from this meter."
    ::= { flowControl 3 }

flowReaderInfoEntry OBJECT-TYPE
    SYNTAX  FlowReaderInfoEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "Information about a particular meter reader."
    INDEX  { flowReaderIndex }
    ::= { flowReaderInfoTable 1 }

FlowReaderInfoEntry ::= SEQUENCE {
    flowReaderIndex            Integer32,
    flowReaderOwner            IpAddress,
    flowReaderLastTime         TimeStamp,
    flowReaderPreviousTime     TimeStamp
    }

flowReaderIndex OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "Selects an entry from the array of meter reader info entries."
    ::= { flowReaderInfoEntry 1 }

flowReaderOwner OBJECT-TYPE
    SYNTAX  IpAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Peer address of this meter reader."
```

```
    ::= { flowReaderInfoEntry 2 }
```

```
flowReaderLastTime OBJECT-TYPE
    SYNTAX  TimeStamp
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Time this meter reader last began a data collection."
    ::= { flowReaderInfoEntry 3 }

flowReaderPreviousTime OBJECT-TYPE
    SYNTAX  TimeStamp
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Time this meter reader began the collection before last."
    ::= { flowReaderInfoEntry 4 }

flowLastReadTime OBJECT-TYPE
    SYNTAX  TimeTicks
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Time last collection of meter data began.  This variable
        will be written by a meter reader as the first step in reading
        flow data.  The meter will set its LastTime value to uptime
        and set its PreviousTime value to the old LastTime.  This
        allows the meter to recover flows which have been inactive
        since PreviousTime, for these have been collected at least
        once.  If the meter fails to write flowLastReadTime, e.g.
        by failing authentication in the meter SNMP write community,
        collection may still proceed but the meter may not be able to
        recover inactive flows."
    ::= { flowControl 4 }

--
-- Control Group:  General Meter Control Variables
--

-- At present the meter only runs a single rule set  - the 'current'
-- one and has a single 'standby' rule set.  In future it may be
-- developed so as to run multiple rule sets simultaneously; that would
-- require a more elaborate set of control variables to allow reliable
-- operation.

flowCurrentRuleSet OBJECT-TYPE
    SYNTAX  INTEGER (0..255)
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Index to the array of rule tables.  Specifies which set of
```

rules is currently being used for accounting by the meter.
When the manager sets this variable the meter will close its

```
        current rule set and start using the new one.  Flows created
        by the old rule set remain in memory, orphaned until their
        data has been read.  Specifying rule set 0 (the empty set)
        stops flow measurement."
    ::= { flowControl 5 }

flowStandbyRuleSet OBJECT-TYPE
    SYNTAX  INTEGER (0..255)
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "Index to the array of rule tables.  After reaching
        HighWaterMark (see below) the meter may switch to using it's
        standby rule set.  For this to be effective the manager should
        have downloaded a standby rule set which uses a coarser
        reporting granularity.  The manager may also need to
        decrease the meter reading interval so that the meter can
        recover flows measured by its normal rule set."
    ::= { flowControl 6 }

flowHighWaterMark OBJECT-TYPE
    SYNTAX  INTEGER (0..100)
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "A value expressed as a percentage, interpreted by the meter
        as an indication of how full the flow table should be before
        it should switch to the standby rule set (if one has been
        specified).  Values of 0% or 100% disable the checking
        represented by this variable."
    ::= { flowControl 7 }

flowFloodMark OBJECT-TYPE
    SYNTAX  INTEGER (0..100)
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
        "A value expressed as a percentage, interpreted by the meter
        as an indication of how full the flow table should be before
        it should take some action to avoid running out of resources
        to handle new flows.  Values of 0% or 100% disable the
        checking represented by this variable."
    ::= { flowControl 8 }

flowInactivityTimeout OBJECT-TYPE
    SYNTAX  Integer32 (1..3600)
    MAX-ACCESS  read-write
    STATUS  current
    DESCRIPTION
```

"The time in seconds since the last packet seen, after
which the flow may be terminated. Note that although a

```
        flow may have been terminated, its data must be collected
        before its memory can be recovered."
    DEFVAL { 600 } -- 10 minutes
    ::= { flowControl 9 }

flowActiveFlows OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The numbers of flows which are currently in use, i.e. have
        been active since the last collection."
    ::= { flowControl 10 }

flowMaxFlows OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The maximum number of flows allowed in the meter's
        flow table.  At present this is determined when the meter
        is first started up."
    ::= { flowControl 11 }


--
-- The Flow Table
--

-- This is a table kept by a meter, with one flow data entry for every
-- flow being measured.  Each flow data entry stores the attribute
-- values for a traffic flow.  Details of flows and their attributes
-- are given in the 'Traffic Flow Measurement: Architecture'
-- document [9].

-- From time to time a meter reader may sweep the flow table so as read
-- counts.  To reduce the number of SNMP requests required to do this,
-- two further tables provide alternative windows into the flow table.
-- The Activity Table allows a meter reader to find all the flows which
-- were active at or after a specified time, and the Column Activity
-- Table allows a meter reader to retrieve part of a column from the
-- flow table.  Note that it is not sensible for the meter to keep
-- the Activity Table in LastActiveTime order, since that would result
-- in very active flows being counted many times during the same
-- collection.

-- This scheme allows multiple meter readers to independently use the
-- same meter; the meter readers do not have to be synchronised and
-- they may use different collection intervals.
```

```
flowDataTable OBJECT-TYPE
```

```
    SYNTAX  SEQUENCE OF FlowDataEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The list of all flows being measured."
    ::= { flowData 1 }

flowDataEntry OBJECT-TYPE
    SYNTAX  FlowDataEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The flow data record for a particular flow."
    INDEX { flowDataIndex }
    ::= { flowDataTable 1 }

FlowDataEntry ::= SEQUENCE {
    flowDataIndex                   Integer32,
    flowDataStatus                  INTEGER,

    flowDataSourceInterface         Integer32,    -- Source Address
    flowDataSourceAdjacentType      AddressType,
    flowDataSourceAdjacentAddress   AdjacentAddress,
    flowDataSourceAdjacentMask      AdjacentAddress,
    flowDataSourcePeerType          AddressType,
    flowDataSourcePeerAddress       PeerAddress,
    flowDataSourcePeerMask          PeerAddress,
    flowDataSourceTransType         INTEGER,
    flowDataSourceTransAddress      TransportAddress,
    flowDataSourceTransMask         TransportAddress,

    flowDataDestInterface           Integer32,    -- Dest Address
    flowDataDestAdjacentType        AddressType,
    flowDataDestAdjacentAddress     AdjacentAddress,
    flowDataDestAdjacentMask        AdjacentAddress,
    flowDataDestPeerType            AddressType,
    flowDataDestPeerAddress         PeerAddress,
    flowDataDestPeerMask            PeerAddress,
    flowDataDestTransType           INTEGER,
    flowDataDestTransAddress        TransportAddress,
    flowDataDestTransMask           TransportAddress,

    flowDataPDUScale                INTEGER,      -- Rule Set
    flowDataOctetScale              INTEGER,
    flowDataRuleSet                 INTEGER,

    flowDataToOctets                Counter32,    -- Source->Dest
    flowDataToPDUs                  Counter32,
    flowDataFromOctets              Counter32,    -- Dest->Source
```

```
    flowDataFromPDUs              Counter32,
    flowDataFirstTime             TimeTicks,    -- Activity times
```

```
    flowDataLastActiveTime          TimeTicks,

    flowDataSourceSubscriberID      OCTET STRING,
    flowDataDestSubscriberID        OCTET STRING,
    flowDataSessionID               OCTET STRING,

    flowDataSourceClass             INTEGER,
    flowDataDestClass               INTEGER,
    flowDataClass                   INTEGER,
    flowDataSourceKind              INTEGER,
    flowDataDestKind                INTEGER,
    flowDataKind                    INTEGER
    }

flowDataIndex OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Value of this flow data record's index within the meter's
        flow table.
        The value 0 should be used as an initial value for SNMP
        GetNext requests when performing a serial search of the flow
        table
        The value 1 should NOT be used; it is reserved as a special
        end marker for the flowColumnActivityData variable."
    ::= { flowDataEntry 1 }

flowDataStatus OBJECT-TYPE
    SYNTAX  INTEGER { inactive(1), current(2), idle(3) }
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Status of this flow data record."
    ::= { flowDataEntry 2 }

flowDataSourceInterface OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Index of the interface associated with the source address
        for this flow.  It's value is one of those contained in the
        ifIndex field of the meter's interfaces table."
    ::= { flowDataEntry 3 }

flowDataSourceAdjacentType OBJECT-TYPE
    SYNTAX  AddressType
    MAX-ACCESS  read-only
```

```
    STATUS  current
    DESCRIPTION
```

```
        "Adjacent address type of the source for this flow.  If
        accounting is being performed at the network level the
        adjacent address will probably be an 802 MAC address, and
        the adjacent address type will indicate the medium type."
    ::= { flowDataEntry 4 }

flowDataSourceAdjacentAddress OBJECT-TYPE
    SYNTAX  AdjacentAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Address of the adjacent device on the path for the source
        for this flow."
    ::= { flowDataEntry 5 }

flowDataSourceAdjacentMask OBJECT-TYPE
    SYNTAX  AdjacentAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "1-bits in this mask indicate which bits must match when
        comparing the adjacent source address for this flow."
    ::= { flowDataEntry 6 }

flowDataSourcePeerType OBJECT-TYPE
    SYNTAX  AddressType
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Peer address type of the source for this flow."
    ::= { flowDataEntry 7 }

flowDataSourcePeerAddress OBJECT-TYPE
    SYNTAX  PeerAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Address of the peer device for the source of this flow."
    ::= { flowDataEntry 8 }

flowDataSourcePeerMask OBJECT-TYPE
    SYNTAX  PeerAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "1-bits in this mask indicate which bits must match when
        comparing the source peer address for this flow."
    ::= { flowDataEntry 9 }
```

```
flowDataSourceTransType OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
```

```
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Transport address type of the source for this flow.  The
        value of this attribute will depend on the peer address type."
    ::= { flowDataEntry 10 }

flowDataSourceTransAddress OBJECT-TYPE
    SYNTAX  TransportAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Transport address for the source of this flow."
    ::= { flowDataEntry 11 }

flowDataSourceTransMask OBJECT-TYPE
    SYNTAX  TransportAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "1-bits in this mask indicate which bits must match when
        comparing the transport source address for this flow."
    ::= { flowDataEntry 12 }

flowDataDestInterface OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Index of the interface associated with the dest address for
        this flow.  This value is one of the values contained in the
        ifIndex field of the interfaces table."
    ::= { flowDataEntry 13 }

flowDataDestAdjacentType OBJECT-TYPE
    SYNTAX  AddressType
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Adjacent address type of the destination for this flow."
    ::= { flowDataEntry 14 }

flowDataDestAdjacentAddress OBJECT-TYPE
    SYNTAX  AdjacentAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Address of the adjacent device on the path for the
        destination for this flow."
```

```
    ::= { flowDataEntry 15 }
```

```
flowDataDestAdjacentMask OBJECT-TYPE
    SYNTAX  AdjacentAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "1-bits in this mask indicate which bits must match when
        comparing the adjacent dest address for this flow."
    ::= { flowDataEntry 16 }

flowDataDestPeerType OBJECT-TYPE
    SYNTAX  AddressType
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Peer address type of the destination for this flow."
    ::= { flowDataEntry 17 }

flowDataDestPeerAddress OBJECT-TYPE
    SYNTAX  PeerAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Address of the peer device for the destination of this flow."
    ::= { flowDataEntry 18 }

flowDataDestPeerMask OBJECT-TYPE
    SYNTAX  PeerAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "1-bits in this mask indicate which bits must match when
        comparing the dest peer type for this flow."
    ::= { flowDataEntry 19 }

flowDataDestTransType OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Transport address type of the destination for this flow.  The
        value of this attribute will depend on the peer address type."
    ::= { flowDataEntry 20 }

flowDataDestTransAddress OBJECT-TYPE
    SYNTAX  TransportAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Transport address for the destination of this flow."
```

```
    ::= { flowDataEntry 21 }
```

```
flowDataDestTransMask OBJECT-TYPE
    SYNTAX  TransportAddress
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "1-bits in this mask indicate which bits must match when
        comparing the transport destination address for this flow."
    ::= { flowDataEntry 22 }

flowDataPDUScale OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The scale factor applied to this particular flow.  Indicates
        the number of bits the PDU counter values should be moved left
        to obtain the actual values."
    ::= { flowDataEntry 23 }

flowDataOctetScale OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The scale factor applied to this particular flow.  Indicates
        the number of bits the octet counter values should be moved
        left to obtain the actual values."
    ::= { flowDataEntry 24 }

flowDataRuleSet OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The RuleSet number of the rule set which created this flow."
    ::= { flowDataEntry 25 }

flowDataToOctets OBJECT-TYPE
    SYNTAX  Counter32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The count of octets flowing from source to dest address and
        being delivered to the protocol level being metered.  In the
        case of IP this would count the number of octets delivered to
        the IP level."
    ::= { flowDataEntry 26 }

flowDataToPDUs OBJECT-TYPE
```

```
    SYNTAX  Counter32
    MAX-ACCESS  read-only
```

```
    STATUS  current
    DESCRIPTION
        "The count of protocol packets flowing from source to dest
        address and being delivered to the protocol level being
        metered.  In the case of IP, for example, this would count the
        IP packets delivered to the IP protocol level."
    ::= { flowDataEntry 27 }

flowDataFromOctets OBJECT-TYPE
    SYNTAX  Counter32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The count of octets flowing from dest to source address and
        being delivered to the protocol level being metered."
    ::= { flowDataEntry 28 }

flowDataFromPDUs OBJECT-TYPE
    SYNTAX  Counter32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The count of protocol packets flowing from dest to source
        address and being delivered to the protocol level being
        metered.  In the case of IP, for example, this would count
        the IP packets delivered to the IP protocol level."
    ::= { flowDataEntry 29 }

flowDataFirstTime OBJECT-TYPE
    SYNTAX  TimeTicks
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The time at which this flow was first entered in the table"
    ::= { flowDataEntry 30 }

flowDataLastActiveTime OBJECT-TYPE
    SYNTAX  TimeTicks
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "The last time this flow had activity, i.e. the time of
        arrival of the most recent PDU belonging to this flow."
    ::= { flowDataEntry 31 }

flowDataSourceSubscriberID OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (4..20))
    MAX-ACCESS  read-only
    STATUS  current
```

```
    DESCRIPTION
        "Subscriber ID associated with the source address for this
```

```
        flow."
    ::= { flowDataEntry 32 }

flowDataDestSubscriberID OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (4..20))
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Subscriber ID associated with the dest address for this
        flow."
    ::= { flowDataEntry 33 }

flowDataSessionID OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (4..10))
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Session ID for this flow.  Such an ID might be allocated
        by a network access server to distinguish a series of sessions
        between the same pair of addresses, which would otherwise
        appear to be parts of the same accounting flow."
    ::= { flowDataEntry 34 }

flowDataSourceClass OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Source class for this flow.  Determined by the rules, set by
        a PushRule action when this flow was entered in the table."
    ::= { flowDataEntry 35 }

flowDataDestClass OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Destination class for this flow.  Determined by the rules, set
        by a PushRule action when this flow was entered in the table."
    ::= { flowDataEntry 36 }

flowDataClass OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Class for this flow.  Determined by the rules, set by a
        PushRule action when this flow was entered in the table."
    ::= { flowDataEntry 37 }
```

```
flowDataSourceKind OBJECT-TYPE
```

```
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Source kind for this flow.  Determined by the rules, set by
        a PushRule action when this flow was entered in the table."
    ::= { flowDataEntry 38 }

flowDataDestKind OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Destination kind for this flow.  Determined by the rules, set
        by a PushRule action when this flow was entered in the table."
    ::= { flowDataEntry 39 }

flowDataKind OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Class for this flow.  Determined by the rules, set by a
        PushRule action when this flow was entered in the table."
    ::= { flowDataEntry 40 }


--
-- The Activity Table
--

flowActivityTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FlowActivityEntry
    MAX-ACCESS  not-accessible
    STATUS   current
    DESCRIPTION
        "Index into the Flow Table.  This is the 'Collect Index'
        described in the 'Traffic Flow Measurement: Architecture'
        document [9].  It allows a meter reader to retrieve a list
        containing the flow table indexess for all flows wich were
        last active at or after a given time."
    ::= { flowData 2 }

flowActivityEntry OBJECT-TYPE
    SYNTAX  FlowActivityEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The Activity Entry for a particular activity time and flow."
```

```
    INDEX { flowActivityTime, flowActivityIndex }
    ::= { flowActivityTable 1 }
```

```
FlowActivityEntry ::= SEQUENCE {
    flowActivityTime            TimeFilter,
    flowActivityIndex           Integer32
    }

flowActivityTime OBJECT-TYPE
    SYNTAX  TimeFilter
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "This variable is a copy of flowDataLastActiveTime in the
        flow data record identified by the flowActivityIndex value
        of this flowActivityTable entry."
    ::= { flowActivityEntry 1 }

flowActivityIndex OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Index of a flow table entry which was active at or after a
        specified flowActivity time.  This index may be used to
        retrieve attribute values for that flow data record."
    ::= { flowActivityEntry 2 }


--
-- The Activity Column Table
--

flowColumnActivityTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FlowColumnActivityEntry
    MAX-ACCESS  not-accessible
    STATUS   current
    DESCRIPTION
        "Index into the Flow Table.  Allows a meter reader to retrieve
        a list containing the flow table indeces of flows which were
        last active at or after a given time, together with the values
        of a specified attribute for each such flow."
    ::= { flowData 3 }

flowColumnActivityEntry OBJECT-TYPE
    SYNTAX  FlowColumnActivityEntry
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The Column Activity Entry for a particular attribute,
        activity time and flow."
```

```
    INDEX { flowColumnActivityAttribute, flowColumnActivityTime,
            flowColumnActivityIndex }
```

```
    ::= { flowColumnActivityTable 1 }

FlowColumnActivityEntry ::= SEQUENCE {
    flowColumnActivityAttribute   FlowAttributeNumber,
    flowColumnActivityTime        TimeFilter,
    flowColumnActivityIndex       Integer32,
    flowColumnActivityData        OCTET STRING
    }

flowColumnActivityAttribute OBJECT-TYPE
    SYNTAX  FlowAttributeNumber
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Specifies the attribute for which values are required from
        active flows."
    ::= { flowColumnActivityEntry 1 }

flowColumnActivityTime OBJECT-TYPE
    SYNTAX  TimeFilter
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "This variable is a copy of flowDataLastActiveTime in the
        flow data record identified by the flowColumnActivityIndex
        value of this flowColumnActivityTable entry."
    ::= { flowColumnActivityEntry 2 }

flowColumnActivityIndex OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Index of a flow table entry which was active at or after
        a specified flowColumnActivityTime."
    ::= { flowColumnActivityEntry 3 }

flowColumnActivityData OBJECT-TYPE
    SYNTAX  OCTET STRING (SIZE (5..1000))
    MAX-ACCESS  read-only
    STATUS  current
    DESCRIPTION
        "Collection of attribute data for flows active after
        flowColumnActivityTime.  Within the OCTET STRING is a
        sequence of { flow index, attribute value } pairs, one for
        each active flow.  The end of the sequence is marked by a
        flow index value of 0 if there are no more rows in this
        column, and 1 otherwise.
```

The format of objects inside flowColumnFlowData is as follows.
All numbers are unsigned. Numbers and strings appear with

their high-order bytes leading. Numbers are fixed size, as specified by their SYNTAX in the flow table (above), i.e. one octet for flowAddressType and small constants, and four octets for Counter and Timeticks. Strings are variable-length, with the length given in a single leading octet.

The following is an attempt at an ASN.1 definition of flowColumnActivityData:

```
flowColumnActivityData ::= SEQUENCE {
   RowItems         flowRowItemList,
   EndMarker        INTEGER (0..1)   -- 0 = No more rows
   }
flowRowItemList ::= SEQUENCE OF flowRowItemEntry
flowRowItemEntry ::= SEQUENCE {
   flowRowNumber    INTEGER (1..65535),
   flowDataValue   flowDataType -- Choice depends on attribute
   }
flowDataType ::= CHOICE {
    flowByteValue   INTEGER (1..255),
    flowShortValue  INTEGER (1..65535),
    flowLongValue   Integer32,
    flowStringValue OCTET STRING  -- Length (n) in first byte,
          -- n+1 bytes total length, trailing zeroes truncated
    }"
::= { flowColumnActivityEntry 4 }


--
-- The Rule Table
--

-- This is an array of rule tables; the one in use is selected by
-- CurrentRuleSet.  To change the rule set the manager chooses a set
-- number which is not in use, downloads the new rule set there, then
-- writes the new set number into CurrentRuleSet.  Rule set 1 is the
-- default rule set, used by the meter on start-up.  Several rule sets
-- can be held in a meter so that the manager can change the rules
-- easily, for example with time of day.  Note that the manager may
-- not change the default rule set, nor the rules in the current rule
-- set!  See the 'Traffic Flow Measurement: Architecture' document [9]
-- for details of rules and how they are used.

flowRuleTable OBJECT-TYPE
    SYNTAX  SEQUENCE OF FlowRuleEntry
    MAX-ACCESS  not-accessible
    STATUS   current
    DESCRIPTION
        "Contains all the rule sets which may be used by the meter."
```

```
    ::= { flowRules 1 }
```

```
flowRuleEntry OBJECT-TYPE
    SYNTAX  FlowRuleEntry
    MAX-ACCESS  not-accessible
    STATUS   current
    DESCRIPTION
        "The rule record itself."
    INDEX { flowRuleSet, flowRuleIndex }
    ::= { flowRuleTable 1 }

FlowRuleEntry ::= SEQUENCE {
    flowRuleSet                 INTEGER,
    flowRuleIndex               INTEGER,
    flowRuleSelector            RuleAttributeNumber,
    flowRuleMask                RuleAddress,
    flowRuleMatchedValue        RuleAddress,
    flowRuleAction              ActionNumber,
    flowRuleParameter           Integer32
    }

flowRuleSet OBJECT-TYPE
    SYNTAX  INTEGER (1..255)
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "Selects a rule set from the array of rule sets."
    ::= { flowRuleEntry 1 }

flowRuleIndex OBJECT-TYPE
    SYNTAX  INTEGER (1..65535)
    MAX-ACCESS  not-accessible
    STATUS  current
    DESCRIPTION
        "The index into the Rule table.  N.B: These values will
        often be consecutive, given the fall-through semantics of
        processing the table."
    ::= { flowRuleEntry 2 }

flowRuleSelector OBJECT-TYPE
    SYNTAX  RuleAttributeNumber
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "Defines the source of the value to match.

        null(0) is a special case; null rules always succeed.

        v1(51), v2(52), v3(53), v4(54) and v5(55) select meter
        variables, each of which can hold the name (i.e. selector
        value) of an address attribute.  When one of these is used
```

as a selector, its value specifies the attribute to be tested. Variable values are set by an Assign action."

```
    ::= { flowRuleEntry 3 }

flowRuleMask OBJECT-TYPE
    SYNTAX  RuleAddress
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "The initial mask used to compute the desired value.  If the
        mask is zero the rule's test will always succeed."
    ::= { flowRuleEntry 4 }

flowRuleMatchedValue OBJECT-TYPE
    SYNTAX  RuleAddress
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "The resulting value to be matched for equality.
        Specifically, if the attribute chosen by the flowRuleSelector
        logically ANDed with the mask specified by the flowRuleMask
        equals the value specified in the flowRuleMatchedValue, then
        continue processing the table entry based on the action
        specified by the flowRuleAction entry.  Otherwise, proceed to
        the next entry in the rule table."
    ::= { flowRuleEntry 5 }

flowRuleAction OBJECT-TYPE
    SYNTAX  ActionNumber
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "The action to be taken if this rule's test succeeds, or if
        the meter's 'test' flag is off.  Actions are opcodes for the
        meter's Packet Matching Engine; details are given in the
        'Traffic Flow Measurement: Architecture' document [9]."
    ::= { flowRuleEntry 6 }

flowRuleParameter OBJECT-TYPE
    SYNTAX  Integer32
    MAX-ACCESS  read-create
    STATUS  current
    DESCRIPTION
        "A parameter value providing extra information for the
        rule's action."
    ::= { flowRuleEntry 7 }


--
-- Accounting Meter conformance statement
--
```

flowMIBCompliances

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

flowMIBGroups
    OBJECT IDENTIFIER ::= { flowMIBConformance 2 }

flowControlGroup OBJECT-GROUP
    OBJECTS  {
        flowRuleInfoSize, flowRuleInfoOwner,
            flowRuleInfoTimeStamp, flowRuleInfoStatus,
        flowInterfaceRate,
            flowInterfaceLostPackets,
        flowReaderOwner,
            flowReaderLastTime, flowReaderPreviousTime,
        flowLastReadTime,
            flowCurrentRuleSet,
            flowStandbyRuleSet,
            flowHighWaterMark,
            flowFloodMark,
            flowInactivityTimeout,
            flowActiveFlows,
            flowMaxFlows }
    STATUS  current
    DESCRIPTION
        "The control group defines objects which are used to control
        an accounting meter."
    ::= {flowMIBGroups 1 }

flowDataTableGroup OBJECT-GROUP
    OBJECTS  {
        flowDataIndex,
        flowDataStatus,
        flowDataSourceInterface,
        flowDataSourceAdjacentType,
        flowDataSourceAdjacentAddress, flowDataSourceAdjacentMask,
        flowDataSourcePeerType,
        flowDataSourcePeerAddress, flowDataSourcePeerMask,
        flowDataSourceTransType,
        flowDataSourceTransAddress, flowDataSourceTransMask,
        flowDataDestInterface,
        flowDataDestAdjacentType,
        flowDataDestAdjacentAddress, flowDataDestAdjacentMask,
        flowDataDestPeerType,
        flowDataDestPeerAddress, flowDataDestPeerMask,
        flowDataDestTransType,
        flowDataDestTransAddress, flowDataDestTransMask,
        flowDataRuleSet,
        flowDataToOctets, flowDataToPDUs,
        flowDataFromOctets, flowDataFromPDUs,
        flowDataFirstTime, flowDataLastActiveTime,
```

```
        flowDataSourceClass, flowDataDestClass, flowDataClass,
        flowDataSourceKind, flowDataDestKind, flowDataKind,
```

```
        flowActivityTime, flowActivityIndex
        }
    STATUS  current
    DESCRIPTION
        "The flow table group defines objects which provide the
        structure for the rule table, including the creation time
        and activity time indexes into it.  In addition it defines
        objects which provide a base set of flow attributes for the
        adjacent, peer and transport layers, together with a flow's
        counters and times.  Finally it defines a flow's class and
        kind attributes, which are set by rule actions."
    ::= {flowMIBGroups 2 }

flowDataScaleGroup OBJECT-GROUP
    OBJECTS  {
        flowDataPDUScale, flowDataOctetScale
        }
    STATUS  current
    DESCRIPTION
        "The flow scale group defines objects which specify scale
        factors for counters."
    ::= {flowMIBGroups 3 }

flowDataSubscriberGroup OBJECT-GROUP
    OBJECTS  {
        flowDataSourceSubscriberID, flowDataDestSubscriberID,
        flowDataSessionID
        }
    STATUS  current
    DESCRIPTION
        "The flow subscriber group defines objects which may be used
        to identify the end point(s) of a flow."
    ::= {flowMIBGroups 4 }

flowDataColumnTableGroup OBJECT-GROUP
    OBJECTS  {
        flowColumnActivityAttribute,
        flowColumnActivityTime,
        flowColumnActivityIndex,
        flowColumnActivityData
        }
    STATUS  current
    DESCRIPTION
        "The flow column table group defines objects which can be used
        to collect part of a column of attribute values from the flow
        table."
    ::= {flowMIBGroups 5 }

flowRuleTableGroup OBJECT-GROUP
```

```
    OBJECTS  {
        flowRuleSelector,
```

```
        flowRuleMask, flowRuleMatchedValue,
        flowRuleAction, flowRuleParameter
        }
    STATUS  current
    DESCRIPTION
        "The rule table group defines objects which hold the set(s)
        of rules specifying which traffic flows are to be accounted
        for."
    ::= {flowMIBGroups 6 }

flowMIBCompliance MODULE-COMPLIANCE
    STATUS  current
    DESCRIPTION
        "The compliance statement for a Traffic Flow Meter."
    MODULE
        MANDATORY-GROUPS  {
            flowControlGroup,
            flowDataTableGroup,
            flowRuleTableGroup
            }
    ::= { flowMIBCompliances 1 }

END
```

5 Acknowledgements

An early draft of this document was produced under the auspices of the IETF's Accounting Working Group with assistance from SNMP and SAAG working groups. Particular thanks are due to Jim Barnes, Sig Handelman and Stephen Stibler for their support and their assistance with checking the MIB.

6 References

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

[2] Case J., McCloghrie K., Rose M., and Waldbusser S., "Structure of Management Information for version 2 of the Simple Network Managemenet Protocol," RFC 1442, SNMP Research Inc., Hughes LAN Systems, Dover Beach Consulting, Carnegie Mellon University, April 1993.

[3] Case J., McCloghrie, K., Rose, M., and Waldbusser, S., "Textual Conventions for version 2 of the Simple Network Managemenet Protocol SNMPv2", RFC 1443, SNMP Research Inc., Hughes LAN Systems, Dover Beach Consulting, Carnegie Mellon University, April 1993.

[4] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Conformance Statements for version 2 of the Simple Network Managemenet Protocol (SNMPv2)," RFC 1444, SNMP Research Inc., Hughes LAN Systems, Dover Beach Consulting, Carnegie Mellon University, April 1993.

[5] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework," RFC 1452, SNMP Research Inc., Hughes LAN Systems, Dover Beach Consulting, Carnegie Mellon University, April 1993.

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

[7] 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.

[8] Mills, C., Hirsch, G. and Ruth, G., "Internet Accounting Background," RFC 1272, Bolt Beranek and Newman Inc., Meridian Technology Corporation, November 1991.

[9] Brownlee, N., Mills, C., and Ruth, G., "Traffic Flow Measurement: Architecture," Internet Draft (work in progress), The University of Auckland, Bolt Beranek and Newman Inc., GTE Laboratories, Inc, February 1995.

[10] Case, J., "FDDI Management Information Base," RFC 1285, SNMP Research Incorporated, January 1992.

[11] Waldbusser, S., "Remote Network Monitoring Management Information Base, Version 2," Internet Draft (work in progress).

7 Security Considerations

Security issues are not discussed in this document.

8 Author's Address

Nevil Brownlee
Computer Centre
The University of Auckland

Phone: +64 9 373 7599 x8941
E-mail: n.brownlee @auckland.ac.nz