Network Working Group
Internet Draft
Document: draft-ietf-ippm-reporting-mib-05.txt
Category: Standards Track

E. Stephan/J. Jewitt
France Telecom R&D

February, 2004

IPPM reporting MIB

Status of this Memo

Abstract

This memo defines a portion of the Management Information Base (MIB) designed for use with network management protocols in TCP/IP-based internets.
In particular, this MIB specifies the objects used for managing the results of the IPPM metrics measures, for pushing alarms, and for reporting the measures results.

Table of Contents

1. Introduction

This memo defines a MIB for managing network measurements based upon the IP performance metrics specified by the IPPM Working Group.

The definition of objects in the IPPM MIB are built on notions introduced and discussed in the IPPM Framework document, RFC 2330 [ii].

This memo defines a Management Information Base (MIB), and as such it is intended to be respectful of the "Boilerplate for IETF MIBs" defined in http://www.ops.ietf.org/mib-boilerplate.html.

There are companion documents to the IPPM-REPORTING-MIB both in the Transport Area (See section 2), and in the Operations and Management

Area (See section 3). The reader should be familiar with these documents.

2. The IPPM Framework

The IPPM Framework consists of 3 major components:

A general framework for defining performance metrics, as described in the Framework for IP Performance Metrics, RFC 2330 [2];

A set of standardized metrics which conform to this framework: The IPPM Metrics for Measuring Connectivity, RFC 2678 [iii]; The One-way Delay Metric for IPPM, RFC 2679 [iv]; The One-way Packet Loss Metric for IPPM, RFC 2680 [v]; The Round-trip Delay Metric for IPPM, RFC 2681 [vi];

Emerging metrics that are being specified in respect of this framework.

3. The SNMP Management Framework

The SNMP Management Framework consists of five major components:

An overall architecture, described in RFC 2571 [2].

Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIv1 and described in STD 16, RFC 1155 [3], STD 16, RFC 1212 [4] and RFC 1215 [5]. The second version, called SMIv2, is described in STD 58, RFC 2578 [6], STD 58, RFC 2579 [7] and STD 58, RFC 2580 [8].

Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [9]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [10] and RFC 1906 [11]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [11], RFC 2572 [12] and RFC 2574 [13].

Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [9]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [14].

A set of fundamental applications described in RFC 2573 [15] and the view-based access control mechanism described in RFC 2575 [16].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [17].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIv2 will be converted into textual descriptions in SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name.

The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to refer to the object type.

4. Overview

Although the number of measurement devices that implement IPPM metrics is growing, there is not currently any standardized management interface to manage remotely the measurement of these metrics. This memo defines a Management Information Base for managing the measurement of IPPM metrics.

To permit metrics to be referenced by other MIBs and other protocols, the IPPM WG has defined a registry of the current metrics and a framework for the integration of future metrics in the [IPPM metrics registry].

As the specification of new metrics is a continuous process, this memo defines a framework for the integration of the future standardized metrics.

The IPPM-REPORTING-MIB introduces a framework where each application identifies its measures in an owner namespace. The administrator may grant access to a measure, or set of measures to another owner via view based access control. As a result, one owner may compute aggregated metrics on another ownerÆs network measures.

Different architectures may be used to perform metric measurements, using a control protocol and a test protocol. Different control frameworks are suitable for performing measurements. The memo lists them, while also looking for a way to integrate them with the IPPM-REPORTING-MIB. This section is for informational purposes only, and is intended to help specify the relationship among the test protocol, the control protocol and the IPPM-REPORTING-MIB.

Special care has been taken to provide a reporting mode suitable for control protocols and test protocols. It addresses the need to provide access to results for the applications.

This MIB is intended to handle multiple concurrent sessions by SNMP applications. However, the SNMP requests are not necessarily to be handled explicitly by the measurement devices, but can be sent to middleware performing an aggregation function. This allows for continuous collection of measurements and statistics computation.

4.1. Textual Conventions

Eight types of data are introduced as textual conventions in this document: IppmOwnerString, IppmOwnerIndex, TimeUnit, PacketType, PacketTypeAddress, GMTTimeStamp, IppmStandardMetrics and IppmMetricResultFilter.

4.1.1 IppmOwnerString

This octet string is used to represent the owners of the various measures and reports in the measurement system.

4.1.2 IppmOwnerIndex

This integer identifies an instance of an object in an owner namespace.

4.1.3 TimeUnit

This textual convention is used to indicate a unit of time, ranging from nanosecond, microsecond, millisecond, second, hour, day, and week.

4.1.4 PacketType and PacketTypeAddress

Section 13 of the IPPM framework [2] introduces the generic notion of a "packet of type P", because in some contexts the metric's value depends on the type of the packets involved in the metric. In the definition of a metric, the type P will be explicitly defined, partially defined, or left generic. Measurement of metrics defined with generic type P are made specific when performing actual measurements. It is important that one be conscious of the exact type of traffic being measured.

The standardization of the management of IPPM measures relies on the capability to unambiguously configure the type P of the packets, and the parameters of the protocol suites of the type P.

RMON2 introduced the concept of protocol identifiers. RFC2895 [xxv] specifies a macro for the definition of protocol identifier. The RFC2896 [xxvi] defines the protocol identifiers for different protocol encapsulation trees.

The type P implementation relies on the MACRO PROTOCOL-IDENTIFIER defined for identifying protocol suites in RMON2. It is achieved by defining the PacketType and the PacketTypeAddress as new syntax in SMIv2 TEXTUAL-CONVENTION.

4.1.4.1 Internet addresses

The section 14 of the IPPM framework defines (for the usual case of a unidirectional path through the Internet) the term "Src" and "Dst". "Src" denotes the IP address of the beginning of the path, and "Dst" denotes the IP address of the end.

The section 3 of the RMON PI Reference specifies the Protocol Identifier Encoding rules, which consists briefly in a recursive

length value format. "Src" and "Dst" are protocol identifier parameters. Their values are encoded in separated fields using the

encoding rules of the protocol identifier, but without trailing parameters.

The packet encapsulation defined in an instance of PacketType embeds the format of "Src" and "Dst" and their values. The type and value of these addresses depend on the type P of the packet, IP version 4, IPV6, IP in IP... Both participate in the completion of the packet encoding.

Examples:

RFC2896 defines the protocol identifiers ip and ipip4. Should there be an Internet tunnel end-point of the IP address 192.168.1.1 in the tunnel 128.2.6.7. the PacketType of the source address of the tunnel, Src, is 'ip.ipip4'. The encoding of 'ip.ipip4' using the RFC2895 rules adds a trailer 2.0.0. It means that an instance of this protocol identifier has 2 parameters, which values will be set only when implemented. In the IPPM PacketType context these 2 parameters are provided in Src (or Dst). In the current example the value of Src is "192.168.1.1 128.2.6.7".

4.1.5 GMTTimeStamp

This textual convention defines the time at which an event occurred. It is very similar to the NTP timestamp format except that it represents the time elapsed since January 1st, 2000 instead of January 1st, 1900.

4.1.6 IppmStandardMetrics

Each standard metric is identified in the IPPM-METRICS-REGISTRY under the node rfc in chronological order. This textual convention defines an octet string to permit several metrics to be performed in a single measure.

4.1.7 Report definition

A report consists of sending, or logging, a subset of results of measurements that have been taken over a period of time. The report defines actions that are taken on the measurement results. An action is performed either:

+ For each result
+ On the results corresponding to a measurement cycle
+ On the results available at the measurement completion.

To preserve the scalability of the whole measurement system, it limits:

+ The amount of data sent to the applications

+ The bandwidth consumption for uploading the result
+ The number of alarms sent to the applications
+ The amount of data saved in the point of measure

Metric thresholds (low, high, inband, outband...) may be defined that indicate when measure values should be reported. These values and their associated time may directly impact service availability.

One may also want to report when particular values (i.e. constantly over a threshold) repeatedly occur over a period of time. For example, if one-way-day is constantly over a specified acceptable threshold value for 10 minutes, then the values should be reported.

The combination of IPPM metric results, threshold events, and event filtering provides a very efficient mechanism to report measurement results, events, and alarms.

A report is described using the TEXTUAL-CONVENTION IppmMetricResultFilter. The report setup must not dramatically increase the amount of data needed by the control protocol to setup a measure:

+ A basic report is defined in the object ippmAggrMeasureFilter;
+ More elaborate reports are described using a metric threshold to generate alarms and events.
+ The generation of alarms and reports requires a management station address to which the data will be sent.
+ SLA alarms are described using an events duration threshold.

The TEXTUAL-CONVENTION IppmMetricResultFilter specifies the list of events and actions that are used to create a report.

4.2 Structure of the MIB

The MIB is arranged as follow:

- ippmSystem Group:

 + ippmPointOfMeasureTable;
 + ippmSynchronizationTable;
+ ippmMetricsTable.

- ippmOwners Group:
-
 + ippmOwnersTable;

- ippmHistory Group:
-

```
    +   ippmHistoryTable;

   - ippmNetMeasure Group:
   -
    +   ippmNetMeasureTable;

   - ippmAggrMeasure Group:
   -
   +   ippmAggrMeasureTable.

   - ippmNotifications
```

4.2.1 The ippmSystem Group

The implementation of this group is mandatory.

This group consists of a set of parameters describing the clock synchronization at a particular point of measure over time, as well as the system clock of the IPPM-REPORTING-MIB agent.

The table ippmPointOfMeasureTable describes the points of measure.

The table ippmSynchronisationTable is critical to the implementation, especially to be respectful of the Section 6.3. of the IPPM Framework, which states that
"Those who develop such measurement methodologies should strive to:
  + Minimize their uncertainties/errors,
  + Understand and document the sources of uncertainty/error, and
  + Quantify the amounts of uncertainty/error."

Consequently the table ippmSynchronisationTable makes these values available to compute reliable statistics.

The table ippmMetricsTable list all the IPPM metrics using the registry order and describes their implementation (unit...).

4.2.2 The ippmOwners Group

This group identifies an owner, or group of owners, that have access to measurements on a probe.

4.2.3 The ippmHistory Group

The results of any given measure are stored in the ippmHistoryTable. The indexing is such that there is an entry in this table for each result of a given measure for a given metric.

4.2.4 The ippmNetMeasure Group

The control protocol registers a description of the existing network measures in the ippmNetMeasureTable.

This group displays the network measures defined by the control protocol. The results are saved in the ippmHistoryTable.

ippmNetMeasureTable is a reflection of the configuration of the network measure.

4.2.5 The ippmAggrMeasure Group

ippmAggrMeasureTable is responsible for the aggregation of results. The aggregated results are saved in the ippmHistoryTable and may be used for higher aggregated measures.

4.2.6 The Notification Group

The Notification group specifies a list of valid notifications. They are used to generate alarms, or reports, to management applications.

4.3 Row identification in an application namespace

IPPM metrics measurement is a distributed task. An owner namespace is defined to avoid the need of polling to determine the next free index, to avoid index collision when 2 applications are looking for a new index as the same time; to increase the speed of the management operations; to reduce bandwidth consumption and to reduce CPU load in the agents and applications.

In a MIB, an object instance identifier is defined by the clause INDEX of the table as a list of objects.

The owner namespace is defined in the INDEX as a couple of 2 objects where the type of first one is IppmOwnerString and the type of the second is IppmOwnerIndex.

The first term of the instance identifier is the name of the owner. The second term is an private index managed by the owner. This index value is unique in an owner namespace. Before the creation of an instance the creator pick up an IppmOwnerIndex value not in use.

This allows the user to choose arbitrary values for the remaining fields of the INDEX clause without checking that the values of these fields exists in the MIB tables. Moreover this allows the owner to use the same instance identifier over a set of IPPM MIB implementations.

Measurements are requested by management applications. An instance of an object managed by a management station is identified by the

management station IppmOwnerString and the private index provided by the MS.

4.4 Relationship of IPPM REPORTING MIB tables

There is inherently a relationship between various tables in the IPPM REPORTING MIB, and as such, the data integrity must be assured. This relationship is depicted in the following examples.

4.4.1 Relationship between the Owners Table and the aggregated measure table

The owners table contains the list of "owners" that can create and activate remotely aggregated measures in an IPPM agent, or read the existing network measures.

It is recommended to make use of "view based access control" in order to restrict access to this table. For example, the master user "administrator" may be given "write" privileges on the ippmOwnersTable, whereas all others are restricted to "read" access. The user "administrator" can then setup the list of other users that have access to measures.

There must be at least 1 owner in the owners' table. This owner may be either setup by default by the IPPM agent, or configured as stated above.

An owner may have multiple corresponding entries in the network and aggregated measure tables. Each entry in a measure table is associated with one, and only one, entry in the owners' table. That is to say, that a defined measure may NOT have multiple owners.

Thus, we have a 1:N relationship between the owners' table and a measure table.

4.4.2 Relationship between the Network Measure Table and the Aggregated Measure Table

The network measure table is read-only, thus entries in this table must be populated by the agent upon startup.
The agent could potentially read a database that contains network measures configured by a 3rd party proprietary management system that directly interacts with the points of measure. However, the "owner" of the measure must be defined in the owners table. It may be either configured directly, or exported to the agent by the external measurement tool.

The aggregated measure table allows for an "owner" to create aggregated measures (such as average, minimum, maximum...) on

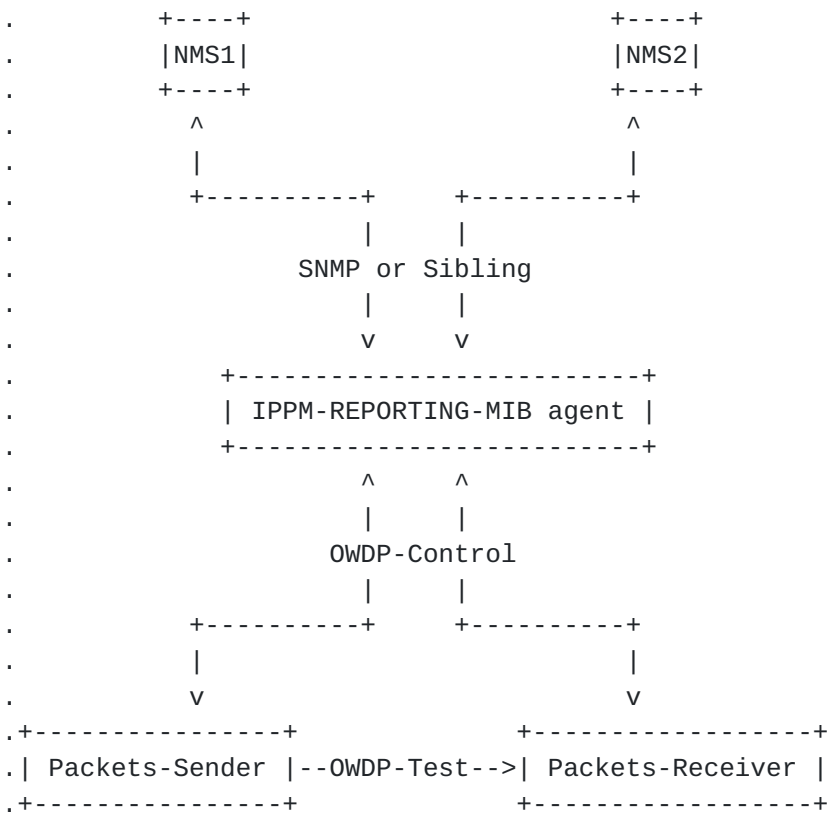
existing measures. An owner may even create aggregated measures on network measures that are owned by other owners. However, it is

recommended to use view based access control to grant access of network measures to other owners in the system.

5 Measurement architectures

There are three main measurement architectures.

5.1 Proxy architecture



In this architecture, the different NMSÆs query the IPPM-REPORTING-MIB agent for measurements. The agent controls whether the NMS is granted access to perform the measure requested. Each NMS may access the results of its measurements in the IPPM-REPORTING-MIB history table.

The measurement setup/teardown and the data collection are done using the control protocol and the test protocol.

In this mode the NMS does not depend on the control protocol nor on the test protocol. The entities involved in the measurement do not need to implement the IPPM-REPORTING-MIB nor SNMP. This mode allows for lightweight implementation in the point of measure, and also for heterogeneous control protocols to coexist.

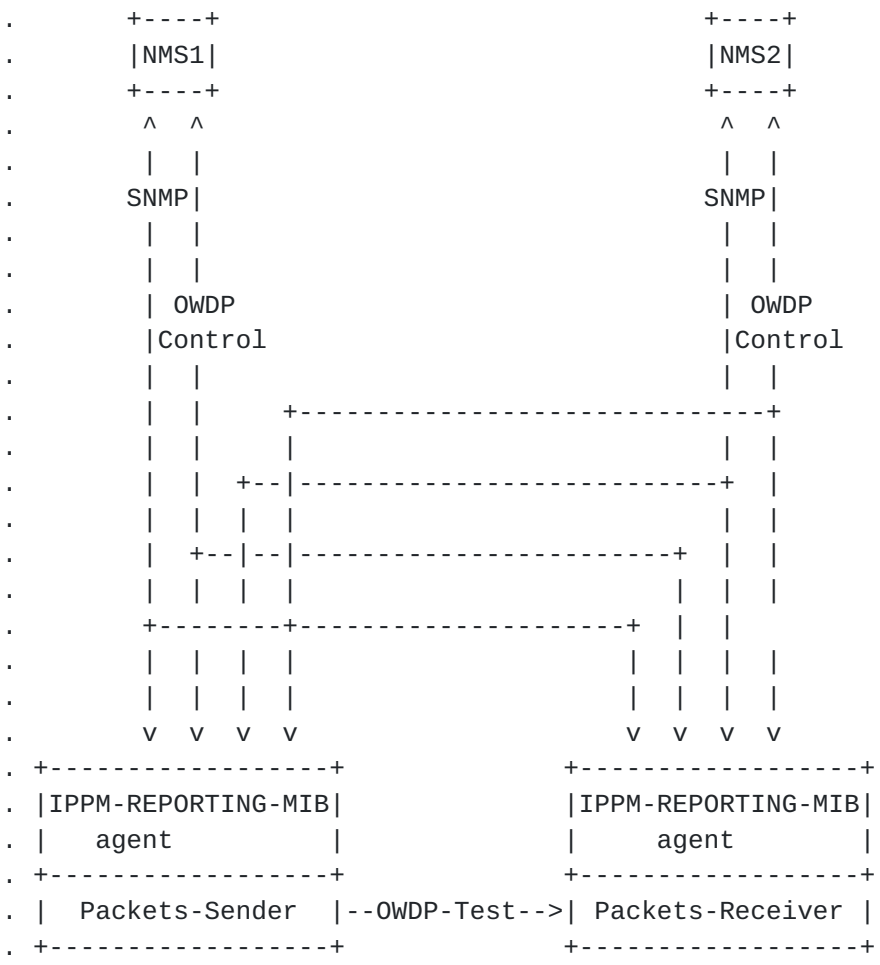
Finally, the proxy is a checkpoint where measurement activity may be

logged, and where access to measurement setups may be tightly

controlled. Thus, it provides a reliable architecture to manage the security of a measurement system.

5.2 Reporting architecture

In this architecture the SNMP protocol is only used to read the results of the measurements in the IPPM-REPORTING-MIB History Table, and also to inform the NMS that an event has occurred.



The activation of a measure by the control protocol or the test protocol creates a measure in the IPPM-REPORTING-MIB Network Measure table. The table in question may be not accessible by SNMP. In this case, a list of the measure identifiers (owner, index) is handled by the measurement software.

Each timestamped result of the measure is logged in the IPPM-REPORTING-MIB History table in order to allow read access to the NMSÆs and event handling.

On completion, the measurement results are managed according to the measure setup:

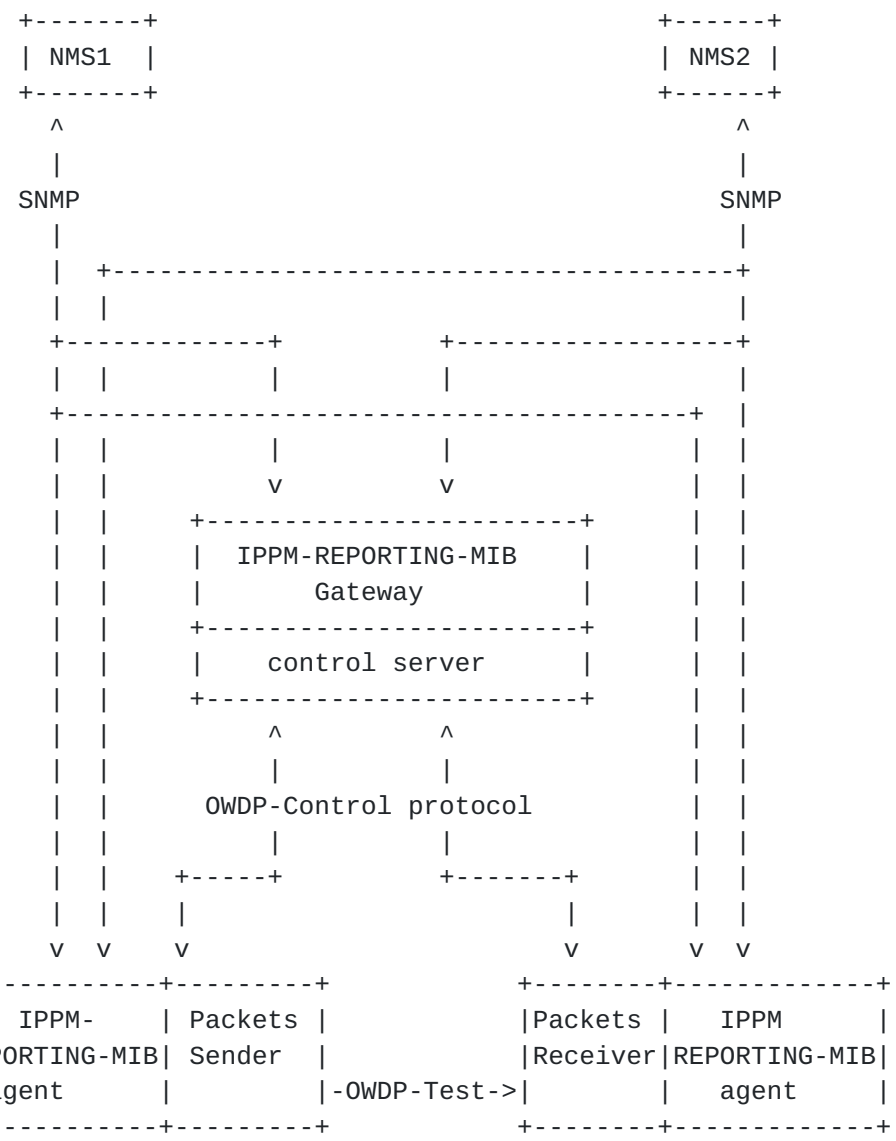
+ The results may be sent to an NMS;

+ They may be dropped from the IPPM-REPORTING-MIB History table.

In this mode, it is recommended to use an SNMPv2 Inform PDU to send reporting events because it ensures that the entire block of the result is received. There is no control using SNMP Trap PDU.

5.3 Gateway architecture

The gateway architecture combines the proxy mode and the reporting mode.



The NMS measurement queries are registered in the IPPM-REPORTING-MIB gateway and performed by the control and the test protocol. The NMS directly consults the result in the corresponding IPPM REPORTING MIB agent of the points of measure.

5.4 Security

The proxy mode provides flexibility and control of the access to the points of measure, while allowing lightweight control protocol and

test protocol implementations in the points of measure. Different security rules may be applied to the NMS domain and to measurement system domains.

The reporting mode has 2 security domains:

+ The control of the measurement setup relies on the control and the test protocol security mechanisms;
+ The control of access to the results depends on the SNMP security mechanisms such as community strings, but may also be restricted using VACM for customized access.

The gateway mode security relies on the security of the proxy mode and of the reporting mode.

6 Reporting mode integration

The IPPM-REPORTING-MIB standardizes the parameters that:

+ Define the configuration of the IPPM metric measures;
+ Define the format of the results of the measure;
+ Define the report of the IPPM metric measure results.

It introduces the concept of owner namespace to allow for fast configuration and reporting across multiple points of measurement.

A measure is a distributed object describing a task to be performed by the control and the test protocols. A measure is identified by its owner and its owner index. This identifier is the same in all the points of measure. As the owner chooses the index, there is no need for negotiation between the NMS and the points of measure before activating the measure.

A measure is primarily defined by its identifier, the metrics to measure, the description of the end point addresses and the description of the scheduling of the measure.

The description of the measure is distributed to the points of measure involved. The distribution may not be synchronized.

6.1 Integration

The integration of the IPPM-REPORTING-MIB, and the test and control protocols consists in pushing the network measure setup/teardown parameters and the result values from the measurement software to the IPPM-REPORTING-MIB agent.

6.2 Setup of the network measure table

The measurement system updates the MIB on creation of a network measure.

6.3 Setup of the aggregated measure table

There are 2 ways to setup an aggregated measure:

The measurement system updates the MIB on creation of an aggregated measure;
An SNMP application creates an aggregated measure.

6.4 Updating the history of the MIB

Results have to be written by the measurement task in the agent implementing the IPPM REPORTING MIB.

Adding the results of a measurement consists in the transfer of the result from the measurement software to the SNMP agent. The protocol that provides the result may be the control protocol, or the test protocol, or another mechanism.

6.5 Default value

The default values correspond to IP version 4.

7 Definition

```
IPPM-REPORTING-MIB DEFINITIONS ::= BEGIN

IMPORTS
MODULE-IDENTITY,
NOTIFICATION-TYPE,
OBJECT-TYPE,
experimental ,Integer32, zeroDotZero, Counter64, Unsigned32
    FROM SNMPv2-SMI
--
-- ippm
--      FROM IPPM-REGISTRY
--
InetAddressType,
InetAddress
    FROM INET-ADDRESS-MIB
SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB
RowStatus,
StorageType,
TEXTUAL-CONVENTION
    FROM SNMPv2-TC
MODULE-COMPLIANCE,
OBJECT-GROUP,
NOTIFICATION-GROUP
    FROM SNMPv2-CONF;

ippmReportingMib MODULE-IDENTITY
   LAST-UPDATED "200402121200Z" --  12 February  2004
   ORGANIZATION "France Telecom - R&D"
```

```
   CONTACT-INFO
       "Emile Stephan
```

France Telecom - R&D
2, Avenue Pierre Marzin
Technopole Anticipa
22307 Lannion Cedex
FRANCE
Tel: + 33 2 96 05 36 10
E-mail: emile.stephan@francetelecom.com

Jessie Jewitt
France Telecom - R&D
801 Gateway Blvd. Suit 500
South San Francisco, CA 94080
Tel : 1 650 875-1524
E-mail : jessie.jewitt@rd.francetelecom.com"

```
   DESCRIPTION
       " 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 specifies the objects used for
       managing the results of the IPPM metrics measurements, alarms and
       reporting of measurement results."

   REVISION "200210181200Z" -- 18 October 2002
   DESCRIPTION
       "General cleanup
       Change 5 tables to read write"

   REVISION "200302141200Z" --  14 February  2003
   DESCRIPTION
       "Modifications based upon feedback from IETF-55"

   REVISION "200306291200Z" --  29 June  2003
   DESCRIPTION
       "Adaptation to VACM, preparation of the final version"

   REVISION "200310241200Z" --  24 October  2003
   DESCRIPTION
       "Modifications based upon feedback from experimental
       implementation."

   REVISION "200402121200Z" --  12 February  2004
   DESCRIPTION
       "Modifications based upon feedback 58th IETF: The report group
       and the corresponding notification are removed."

   ::= { experimental 10001 } -- XXX to be assigned by IANA

ippm           OBJECT IDENTIFIER   ::= { experimental 10000 }
```

```
--
-- TEXTUAL-CONVENTION
```

```
--

IppmOwnerString ::= TEXTUAL-CONVENTION
   STATUS       current
   DESCRIPTION
       "The owner namespace is defined in the INDEX of a table as a
       couple of 2 objects where the type of the first one is
       IppmOwnerString and the type of the second is IppmOwnerIndex.
       IppmOwnerString is an OwnerString which length is limited to 32
       bytes."
       SYNTAX OCTET STRING (SIZE (0..32))

IppmOwnerIndex ::= TEXTUAL-CONVENTION
   STATUS       current
   DESCRIPTION
       "The owner namespace is defined in the INDEX of a table as a
       couple of 2 objects where the type of first one is
       IppmOwnerString and the type of the second is IppmOwnerIndex.
       An object of type IppmOwnerIndex uniquely identifies a row of a
       table inside an owner namespace.
       Inside one namespace several objects of type IppmOwnerIndex
       coexist and share the IppmOwnerIndex range of values to provide a
       unique instance identifier.
       "
       SYNTAX Unsigned32 (1.. 65535)

TimeUnit ::= TEXTUAL-CONVENTION
   STATUS       current
   DESCRIPTION
       "A enumerated list of time units."
   SYNTAX       INTEGER {
       week(1),
       day(2),
       hour(3),
       minute(4),
       second(5),
       millisecond(6),
       microsecond(7),
       nanosecond(8)
   }
--
--

IppmStandardMetrics ::= TEXTUAL-CONVENTION
   STATUS      current
   DESCRIPTION
       " Each standard metric is identified in the IPPM-METRICS-
       REGISTRY under the node rfc in chronological order. In order to
```

allow for several metrics to be calculated in a single measure, there is a need to describe in a bit string the metrics to be measured.

```
       This textual convention defines an octet string that gathers in a
       bit string a sequence of bits. The bit order corresponds to the
       order of the metric identifiers in the registry.
       The first bit of the string has the index 0. The index 1
       corresponds to the first metric of the registry
       (instantaneousUnidirectionalConnectivity ).

       Example:
       One-way-Delay(6) is identified as the leaf number 6 of the node
       rfc of the registry. One-way-Packet-Loss(12) is identified as the
       leaf number 12 of the node
       rfc of the registry. A network measure performing both One-way-
       Delay(6) and One-way-Packet-Loss(12) will be described as
       '0001000001000000'b, '1040'B.
       "
   SYNTAX OCTET STRING (SIZE (1..64))


IppmMetricsRegistryIndex ::= TEXTUAL-CONVENTION
   STATUS      current
   DESCRIPTION
       "IppmMetricsRegistryIndex defines an unambiguous index for each
       standardized metric. It identifies a metric, and as such its
       value is the value of the node of the metric in the IPPM
       registry. This value is the same in any implementation of the
       IPPM REPORTING MIB.

       Example:
       In the IPPM-METRICS-REGISTRY, onewayPacketLossAverage is
       registered as the node 14 of ippmMetricsRegistry.metrics.rfc.
       Consequently the index of the metric onewayPacketLossAverage in
       the IppmMetricsTable will always be '14'. At large an instance,
       which type is IppmMetricsRegistryIndex and which value is '14',
       points to the metric onewayPacketLossAverage."
   SYNTAX Unsigned32 (1.. 65535)



GMTTimeStamp ::= TEXTUAL-CONVENTION
   STATUS       current
   DESCRIPTION
       "The time value at which a measure or an event took place.

       field  octets  contents                  range
       -----  ------  --------                  -----
       1       1-4    second since 1 Jan 1900 0H00*    0..2^31 - 1
       2       5-8    fractional part of the second*   0..2^32 - 1
       * the value is in network-byte order
```

The timestamp format is the NTP timestamp format [RFC 1305].
The reference of time is GMT.

```
        "
    SYNTAX       OCTET STRING (SIZE (8))


PacketType  ::= TEXTUAL-CONVENTION
    STATUS       current
    DESCRIPTION
        "This textual convention is a display string used to describe the
        protocol encapsulation list of a packet, and is used as the value
        of the SYNTAX clause for the type of the Src and Dst of an IPPM
        measure. The RFC2895 specifies a macro named PROTOCOL-IDENTIFIER
        for the definition of protocol identifiers, while its companion
        document, the RFC2896 defines a set of protocol identifiers.

        PacketType is defined as a display string. It consists of a list
        of dot separated protocol names. Each protocol name has been
        previously defined using the macro PROTOCOL-IDENTIFIER of the RFC
        2895.

        Examples:
        The RFC2896 defines the protocol identifiers 'ether2', 'ip',
        'ipip4', 'udp', 'tcp', 'telnet'...

        The PacketType of the source address corresponding to telnet is
        the string 'ip.tcp.telnet'.

        The PacketType of the source address corresponding to UDP packets
        sent in an IP tunnel is the string 'ip.ipip4.udp'.

        Note:
        An IPPM measure is active, so generally a PacketType value does
        not describe the link layer (i.e. ether2...). Valid Internet
        packets are sent from Src to Dst. Then the choice of the link
        layer relies on the Internet stack."
    SYNTAX       OCTET STRING (SIZE (0..512))

PacketTypeAddress ::= TEXTUAL-CONVENTION
    DISPLAY-HINT "255a"
    STATUS       current
    DESCRIPTION
        "This textual convention is a Display string used to describe the
        parameters of the protocol encapsulation list of a packet,
        basically the address.

        PacketTypeAddress is defined as a display string. It consists in
        a list of blank separated addresses that reflect the
        encapsulation of the PacketType. Each parameter in the list
        corresponds to a parameter of a PROTOCOL-IDENTIFIER of the
        PacketType.
```

Example:

```
       The PacketType 'ip.ipip4' has 2 parameters. A valid
       PacketTypeAddress value is '192.168.1.1 128.2.6.7'."
   SYNTAX       OCTET STRING (SIZE (0..512))

IppmMetricResultFilter ::= TEXTUAL-CONVENTION
   STATUS        current
   DESCRIPTION
   "    Given that not all results from a metric measurement are
   pertinent, and that the size of the history must be limited whenever
   possible, the TC IppmMetricResultFilter defines basic filters to
   limit the among of data collected:

   Filter's parameters are the 2 fields ippmAggrMeasureLowThreshold and
   ippmAggrMeasureLowThreshold of the aggregated measure setup.

   A filter determines if the result of the current aggregation has to
   be stored:

       LogInBandValue:
           The value is stored if it is lower than the high threshold of
           the aggregated measure setup and greater than the low
           threshold of of the aggregated measure setup.

       LogOutBandValue:
           The value is stored if it is greater than the high threshold
           of the aggregated measure setup or lower than the low
           threshold of the aggregated measure setup.

       LogAboveValue:
           The value is stored if it is greater than the high threshold
           of the aggregated measure setup.

       LogBelowValue:
           The value is stored if it is lower than the low metric
           threshold field of the aggregated measure setup.

       logUpAndDownValue:
           This filter stores contiguous results that are on opposite
           sides of the up and down metric thresholds:
                A result is stored if it is the first result aggregated:
           If it is greater than the high threshold and lower than the
           low threshold then its value is set to the value of the low
           threshold;

                A result greater than the high threshold is stored if the
           previous result is lower than the low threshold;

                A result lower than the low threshold is stored if the
           previous result is greater than the high threshold;
```

"

```
   SYNTAX       INTEGER {
       logInBandValue(1),
       logOutBandValue(2),
       logAboveValue(3),
       logBelowValue(4),
       logUpAndDownValue(5)
   }


--
-- IPPM  Notifications
--
ippmNotifications OBJECT IDENTIFIER ::= { ippm 0 }

--
-- IPPM  Conformance
--
ippmConformance      OBJECT IDENTIFIER   ::= { ippm 1 }

--
-- IPPM MIB Object definitions
--

ippmSystem           OBJECT IDENTIFIER   ::= { ippmReportingMib 1 }
ippmOwners           OBJECT IDENTIFIER   ::= { ippmReportingMib 2 }
ippmHistory          OBJECT IDENTIFIER   ::= { ippmReportingMib 3 }
ippmNetMeasure       OBJECT IDENTIFIER   ::= { ippmReportingMib 4 }
ippmAggrMeasure      OBJECT IDENTIFIER   ::= { ippmReportingMib 5 }



--
-- ippmSystem  Group
--
--


ippmSystemTime OBJECT-TYPE
   SYNTAX GMTTimeStamp
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The current time of the system running the IPPM REPORTING MIB
       SNMP agent. When the agent is running in proxy mode, it is the
       current time of the proxy agent.
       When the agent is located in the probe, it is the current time of
       the probe agent. "
   ::= { ippmSystem  1 }
```

```
ippmSystemSynchronizationType OBJECT-TYPE
   SYNTAX INTEGER  {
        other(0),
```

```
        ntp(1),
        gps(2),
        cdma(3)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "ippmSystemSynchronizationType describes the mechanism
        used to synchronize the system running  the IPPM REPORTING MIB
        SNMP agent.

        Other(0)
        The synchronization process must be defined
        in the ippmSystemSynchonizationDescription.

        Ntp(1)
        The system is synchronized using the network
        time protocol. The NTP synchronization must be described
        in the ippmSystemSynchonizationDescription.

        Gps(2)
        The system is synchronized using the GPS clocks.

        Cdma(3)
        The system is synchronized using the CDMA clocks."
    ::= { ippmSystem 2 }

ippmSystemSynchronizationDesc OBJECT-TYPE
    SYNTAX SnmpAdminString
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The description of the synchronization process of the system
        running the IPPM REPORTING MIB SNMP agent."
    ::= { ippmSystem 3 }

ippmSystemClockResolution OBJECT-TYPE
    SYNTAX     Unsigned32
    UNITS "Nanoseconds"
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "ippmSystemClockResolution provides the precision of the clock
        used for the measures . The unit is the nanosecond. For example,
        the clock on an old Unix host might advance only once every 10
        msec, and thus have a resolution of 10 msec. So its resolution is
        10000000 nanoseconds and the value of ippmSystemClockResolution
        is 10000000."
```

```
   ::= { ippmSystem 4 }

ippmSystemOperationalStatus OBJECT-TYPE
```

```
   SYNTAX INTEGER  {
        unknown(0),
        up(1),
        down(2)
   }
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
   "This object describes the status of the system running the IPPM
   REPORTING MIB SNMP agent. It does not describe end point measurement
   status.
       unknown(0) means the service is unknown.
       up(1) means the service is operational and available for general
       use.
       down(2) means the agent is not available for use.
       "
   ::= { ippmSystem 5 }

ippmSystemAggregatedMetrics OBJECT-TYPE
   SYNTAX     IppmStandardMetrics
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       " ippmSystemAggregatedMetrics lists the aggregated metrics that
       are performed in the SNMP agent instead of in the point of
       measure."
   ::= { ippmSystem 6 }


ippmSynchronizationTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF IppmSynchronizationEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "This table registers the event related to the synchronization of
       the points of measure. Each event is described in an
       ippmSynchronizationEntry.
       ippmSynchronizationTable is mandatory.
       ippmSynchronizationTable content is read only."
   ::= { ippmSystem 7 }

   ippmSynchronizationEntry OBJECT-TYPE
   SYNTAX     IppmSynchronizationEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "An entry describes a modification of the synchronization
       status."
```

```
   INDEX { ippmPointOfMeasureIndex, ippmSynchronizationIndex }
   ::= { ippmSynchronizationTable 1 }
```

```
IppmSynchronizationEntry ::=
   SEQUENCE {
       ippmSynchronizationIndex              Unsigned32,
       ippmSynchronizationTime               GMTTimeStamp,
       ippmSynchronizationStratum            Unsigned32,
       ippmSynchronizationResolution         Unsigned32
   }

ippmSynchronizationIndex     OBJECT-TYPE
   SYNTAX     Unsigned32 (1 .. 65535)
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "An index that identifies the synchronization events in
       chronological order.
       65535 is an arbitrary size. It is not recommended to keep
       permanently a history of 65535 events."
   ::= { ippmSynchronizationEntry 1 }


ippmSynchronizationTime OBJECT-TYPE
SYNTAX GMTTimeStamp
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The time when the synchronization event occurs."
   ::= { ippmSynchronizationEntry 2 }

ippmSynchronizationStratum OBJECT-TYPE
   SYNTAX     Unsigned32
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The stratum level of the clock computed when the synchronization
       event occurs."
   ::= { ippmSynchronizationEntry 3 }

ippmSynchronizationResolution OBJECT-TYPE
   SYNTAX     Unsigned32
   UNITS      "Nanoseconds"
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The new time resolution computed after the synchronization event
       occurred."
   ::= { ippmSynchronizationEntry 4 }

ippmPointOfMeasureTable OBJECT-TYPE
```

```
   SYNTAX     SEQUENCE OF IppmPointOfMeasureEntry
   MAX-ACCESS not-accessible
   STATUS     current
```

```
   DESCRIPTION
       " This table is the list of measurement end points available in
       the measurement system.

       Proxy mode:
       It is the list of the measurement end points of the set of probes
       for which the IPPM agent provides an SNMP interface.

       IPPM MIB implemented in a probe:
       It is the list of the measurement end points of the probe.

       The ippmPointOfMeasureTable content is read only. This implies
       that the measurement software handles the table internally

       ippmPointOfMeasureTable is mandatory."
   ::= { ippmSystem 8 }

ippmPointOfMeasureEntry OBJECT-TYPE
   SYNTAX     IppmPointOfMeasureEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       " An entry may be the management address of some middleware in
       charge of the management of a set of probes. It may the
       management address of a probe that contains several line cards.

       An entry describes the capability of a point of measure.

       ippmPointOfMeasureMetrics lists the metrics handles by the point
       of measure."
   INDEX { ippmPointOfMeasureIndex }
   ::= { ippmPointOfMeasureTable 1 }

IppmPointOfMeasureEntry ::= SEQUENCE {
   ippmPointOfMeasureIndex               Unsigned32,
   ippmPointOfMeasureMgmtAddrType        InetAddressType,
   ippmPointOfMeasureMgmtAddress         InetAddress,
   ippmPointOfMeasureTestAddrType     InetAddressType,
   ippmPointOfMeasureTestAddress         InetAddress,
   ippmPointOfMeasureMetrics             IppmStandardMetrics
}



ippmPointOfMeasureIndex OBJECT-TYPE
   SYNTAX Unsigned32 (1 .. 65535)
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
```

"A local index that identifies an entry in the point of measure table."

```
   ::= { ippmPointOfMeasureEntry 1 }

ippmPointOfMeasureMgmtAddrType OBJECT-TYPE
   SYNTAX InetAddressType
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The address type associated with the management address."
   ::= { ippmPointOfMeasureEntry 2 }


ippmPointOfMeasureMgmtAddress OBJECT-TYPE
   SYNTAX InetAddress  (SIZE  (1..128))
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The management address on the point of measure"
   ::= { ippmPointOfMeasureEntry 3 }


ippmPointOfMeasureTestAddrType OBJECT-TYPE
   SYNTAX InetAddressType
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Defines the address type of the measurement interface of the
       point of measure."
   ::= { ippmPointOfMeasureEntry 4 }

ippmPointOfMeasureTestAddress OBJECT-TYPE
   SYNTAX InetAddress
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Specifies the address of the measurement interface for the point
       of measure."
   ::= { ippmPointOfMeasureEntry 5}

ippmPointOfMeasureMetrics OBJECT-TYPE
   SYNTAX     IppmStandardMetrics
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       " ippmPointOfMeasureMetrics lists the metrics this point of
       measure implements."
   ::= { ippmPointOfMeasureEntry 6 }


ippmMetricsTable OBJECT-TYPE
```

```
    SYNTAX     SEQUENCE OF IppmMetricsEntry
    MAX-ACCESS not-accessible
```

```
   STATUS     current
   DESCRIPTION
       "This table is mandatory. It describes the current
       implementation. Each IPPM standardized metric must be described
       in the table.
       ippmMetricsTable content is read only."
   ::= { ippmSystem 9 }

ippmMetricsEntry OBJECT-TYPE
   SYNTAX     IppmMetricsEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "An entry describes the static capabilities of a metric
       implementation."
   INDEX { ippmMetricsIndex }
   ::= { ippmMetricsTable 1 }

IppmMetricsEntry ::=
SEQUENCE {
   ippmMetricsIndex           IppmMetricsRegistryIndex,
   ippmMetricsType            INTEGER,
   ippmMetricsUnit            INTEGER,
   ippmMetricsDescription     SnmpAdminString
}

ippmMetricsIndex OBJECT-TYPE
   SYNTAX IppmMetricsRegistryIndex
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "ippmMetricsIndex defines an unambiguous index for each
       standardized metric. It identifies a metric, and as such its
       value is the value of the node of the metric in the IPPM
       registry. This value is the same in any implementation of the
       IPPM REPORTING MIB.

       Example:
       In the IPPM-METRICS-REGISTRY, onewayPacketLossAverage is
       registered as the node 14 of ippmMetricsRegistry.metrics.rfc.
       Consequently the index of the metric onewayPacketLossAverage in
       the IppmMetricsTable will always be '14'"
   ::= { ippmMetricsEntry 1 }


ippmMetricsType OBJECT-TYPE
   SYNTAX INTEGER {
   network(0),
```

```
   aggregated(1)
   }
   MAX-ACCESS read-only
```

```
   STATUS     current
   DESCRIPTION
       "Indicates the metric type, whether it is network or aggregated"
   ::= { ippmMetricsEntry 2 }


ippmMetricsUnit OBJECT-TYPE
   SYNTAX INTEGER {
        noUnit(0),
        second(1),
        millisecond(2),
        microsecond(3),
        nanosecond(4),
        percentage(5),
        packet(6),
        byte(7),
        kilobyte(8),
        megabyte(9)
        }
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The unit used in the current entity for the results of the
       measurement of this metric."
   ::= { ippmMetricsEntry 3 }

ippmMetricsDescription OBJECT-TYPE
   SYNTAX SnmpAdminString
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "A textual description of the metric implementation following the
       exact name of this metric in the registry. For example:
       oneWayDelay: OWD Metric ."
   ::= { ippmMetricsEntry 4 }

--
-- ippmOwners Group
--
-- The ippmOwners objects are responsible for managing
-- the owners access to the measurements.
--
--
ippmOwnersTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF IppmOwnersEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
```

"A management entity wishing to access or aggregate remote Ippm measurements in an agent must previously be registered in the ippmOwnersTable. This table is read-create and contains at least the owner 'monitor'."

```
   ::= { ippmOwners 1 }

ippmOwnersEntry OBJECT-TYPE
   SYNTAX     IppmOwnersEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "The description of the resources granted to an SNMP application.

       For example, an instance of ippmOwnersOwner with an
       IppmOwnerString 'acme', which represents the 14th owner created
       in ippmOwnersTable would be named ippmOwnersEntryOwner.14.
       "
   INDEX { ippmOwnersOwner }
   ::= { ippmOwnersTable 1 }

IppmOwnersEntry ::= SEQUENCE {
   ippmOwnersOwner              IppmOwnerString,
   ippmOwnersGrantedMetrics     IppmStandardMetrics,
   ippmOwnersQuota              Unsigned32,
   ippmOwnersIpAddressType      InetAddressType,
   ippmOwnersIpAddress          InetAddress,
   ippmOwnersEmail              SnmpAdminString,
   ippmOwnersStatus             RowStatus
}

ippmOwnersOwner OBJECT-TYPE
   SYNTAX     IppmOwnerString
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "The owner described by this entry."
   ::= { ippmOwnersEntry 1 }

ippmOwnersGrantedMetrics OBJECT-TYPE
   SYNTAX     IppmStandardMetrics
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       " Defines the metrics granted to an owner for which measurements
       can be performed."
   ::= { ippmOwnersEntry 2 }

ippmOwnersQuota OBJECT-TYPE
   SYNTAX     Unsigned32
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
```

```
       " The maximum number of records that this owner may have in the
       history table and in the report table."
   ::= { ippmOwnersEntry 3 }
```

```
ippmOwnersIpAddressType OBJECT-TYPE
   SYNTAX     InetAddressType
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "The IP address type of the management entity corresponding to
       this owner.
        InetAddressType is restricted to ipv4(1),ipv6(2)and dns(16). "
   ::= { ippmOwnersEntry 4 }

ippmOwnersIpAddress OBJECT-TYPE
   SYNTAX     InetAddress  (SIZE  (1..128))
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "The IP address of the management entity corresponding to this
       owner. For example, the IP address of the management console used
       to send SNMP requests."
   ::= { ippmOwnersEntry 5 }

ippmOwnersEmail OBJECT-TYPE
   SYNTAX     SnmpAdminString
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "The email address of the management entity corresponding to this
       owner."
   ::= { ippmOwnersEntry 6 }


ippmOwnersStatus OBJECT-TYPE
   SYNTAX     RowStatus
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "The status of this table entry. Once this status is set to
       active, the corresponding entry in the table may not be
       modified."
   ::= { ippmOwnersEntry 7 }


--
-- ippmHistory Group
--
--
-- ippmHistoryTable
--
```

```
ippmHistoryTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF IppmHistoryEntry
   MAX-ACCESS not-accessible
```

```
   STATUS     current
   DESCRIPTION
       "The table containing the measurement results."
   ::= { ippmHistory 1 }


ippmHistoryEntry OBJECT-TYPE
   SYNTAX     IppmHistoryEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "An ippmHistoryEntry entry is one of the results of a measure
       identified by ippmHistoryMeasureOwner, ippmHistoryMeasureIndex,
       ippmHistoryMetricIndex and ippmHistorySequence.

       In the index :

       + ippmHistoryMeasureOwner identifies the owner of the measure;

       + ippmHistoryMeasureIndex identifies the measure in the owner
       namespace;

       + ippmHistoryMetricIndex identifies the metric measured by the
       measure. The metric is described in the corresponding entry of
       the  ippmMetricsTable;

       + ippmHistorySequence is the sequence number of the measurement
       result for an entry in the history table."
   INDEX { ippmHistoryMeasureOwner, ippmHistoryMeasureIndex,
   ippmHistoryMetricIndex, ippmHistorySequence }
   ::= { ippmHistoryTable 1 }

IppmHistoryEntry ::=
   SEQUENCE {
       ippmHistoryMeasureOwner      IppmOwnerString,
       ippmHistoryMeasureIndex      IppmOwnerIndex,
       ippmHistoryMetricIndex       IppmMetricsRegistryIndex,
       ippmHistorySequence          Unsigned32,
       ippmHistoryTimestamp         GMTTimeStamp,
       ippmHistoryValue             Integer32
   }

ippmHistoryMeasureOwner OBJECT-TYPE
   SYNTAX     IppmOwnerString
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "The owner of the measure that produced this result. The measure
       is either an ippmNetMeasure or an ippmAggrMeasure."
```

```
   ::= { ippmHistoryEntry 1 }
```

```
ippmHistoryMeasureIndex OBJECT-TYPE
   SYNTAX IppmOwnerIndex
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "The owner index of the measure that produced this result. The
       measure is either an entry of the ippmNetMeasureTable or of the
       ippmAggrMeasureTable."
   ::= { ippmHistoryEntry 2 }

ippmHistoryMetricIndex OBJECT-TYPE
   SYNTAX IppmMetricsRegistryIndex
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       " ippmHistoryMetricIndex identifies the metric measured by the
       measure. The metric is described in the corresponding entry of
       the ippmMetricsTable."
   ::= { ippmHistoryEntry 3 }



ippmHistorySequence OBJECT-TYPE
       SYNTAX Unsigned32 (0..4294967295)
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
       "ippmHistorySequence is the sequence number of the measurement
       results for a metric.

       Network metrics:
       It's the sequence number of a measurement packet. Typically, it
       identifies the order of the packet in the stream of packets sent
       by the source.

       Aggregated metrics:
       It is the sequence order of the aggregation computed."
   ::= { ippmHistoryEntry 4 }



ippmHistoryTimestamp OBJECT-TYPE
   SYNTAX GMTTimeStamp
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The timestamp when the measurement occurred."
   ::= { ippmHistoryEntry 5 }
```

```
ippmHistoryValue OBJECT-TYPE
   SYNTAX Integer32
```

```
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The observed value of the measurement."
   ::= { ippmHistoryEntry 6 }

ippmHistoryPathToResults   OBJECT-TYPE
   SYNTAX SnmpAdminString
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "It is typically an URL describing the file location where bulk
       results are logged."
   ::= { ippmHistory 2 }



--
-- ippmNetMeasure Group
--


--
-- ippmNetMeasureTable
--
--

ippmNetMeasureTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF IppmNetMeasureEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "An entry is a measurement that performs network measures and
       provides results.
       It performs several metric measurements per packet exchange. Each
       step of a measure produces a singleton result per metric. The
       time of the measurement and the value of the metric are saved in
       the ippmHistoryTable."
   ::= { ippmNetMeasure 1 }

ippmNetMeasureEntry OBJECT-TYPE
   SYNTAX     IppmNetMeasureEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       " The IppmNetMeasureTable is mandatory, and its content is read
       only. It means that the measurement software handles the table
       internally. The setup of the network measure is not permitted
       through the IPPM REPORTING MIB. As an example, OWAP provides a
```

setup protocol to setup and tear down networks measures.

```
       The ippmNetMeasureMetrics is set to a list of metrics to be
       computed from the same raw packet exchange. Each step of
       measurement delivers a singleton per metric. Results are
       timestamped and saved in the ippmHistoryTable.

       One may create aggregated measures by using the results of
       network measures."

   INDEX { ippmNetMeasureOwner, ippmNetMeasureIndex }
   ::= { ippmNetMeasureTable 1 }

IppmNetMeasureEntry ::= SEQUENCE {
   ippmNetMeasureOwner               IppmOwnerString,
   ippmNetMeasureIndex               IppmOwnerIndex,
   ippmNetMeasureName                SnmpAdminString,
   ippmNetMeasureMetrics             IppmStandardMetrics,
   ippmNetMeasureBeginTime           GMTTimeStamp,
   ippmNetMeasureCollectionRateUnit  TimeUnit,
   ippmNetMeasureCollectionRate      Unsigned32,
   ippmNetMeasureDurationUnit        TimeUnit,
   ippmNetMeasureDuration            Unsigned32,
   ippmNetMeasureHistorySize         Unsigned32,
   ippmNetMeasureFailureMgmtMode     INTEGER,
   ippmNetMeasureResultsMgmt         INTEGER,
   ippmNetMeasureSrcPacketType       PacketType,
   ippmNetMeasureSrc                 PacketTypeAddress,
   ippmNetMeasureDstPacketType       PacketType,
   ippmNetMeasureDst                 PacketTypeAddress,
   ippmNetMeasureTxMode              INTEGER,
   ippmNetMeasureTxPacketRateUnit    TimeUnit,
   ippmNetMeasureTxPacketRate        Unsigned32,
   ippmNetMeasureMedOrBurstSize      Unsigned32,
   ippmNetMeasureDevOrIntBurstSize   Unsigned32,
   ippmNetMeasureLossTimeout         Unsigned32,
   ippmNetMeasureL3PacketSize        Unsigned32,
   ippmNetMeasureDataPattern         OCTET STRING,
   ippmNetMeasureTotalPktsRecv       Counter64,
   ippmNetMeasureLastUpdate          GMTTimeStamp,
   ippmNetMeasureOperState           INTEGER
}


ippmNetMeasureOwner OBJECT-TYPE
   SYNTAX     IppmOwnerString
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "The owner of the network measure."
```

```
   ::= { ippmNetMeasureEntry 1 }

ippmNetMeasureIndex OBJECT-TYPE
   SYNTAX IppmOwnerIndex
```

```
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "The owner index of the network measure."
   ::= { ippmNetMeasureEntry 2 }

ippmNetMeasureName OBJECT-TYPE
   SYNTAX SnmpAdminString
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The name of the metric instance(s) as defined in
       ippmNetMeasureMetrics. It illustrates the specificity of the
       metric(s) and includes the metric(s) and the PacketType.

       Example:
       IP-TCP-HTTP-One-way-Delay: free text "
   ::= { ippmNetMeasureEntry 3 }



ippmNetMeasureMetrics OBJECT-TYPE
   SYNTAX IppmStandardMetrics
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "ippmNetMeasureMetrics defines the metrics to compute within this
       measure. Only network metrics of the same type are allowed in
       this field (e.g. poisson-based metrics and periodic-based metrics
       are incompatibles, while one-way delay and packet loss are
       generally processed simultaneously: a very bad delay is
       potentially a very good packet loss).

       Results are saved in the ippmHistoryTable. Results of a metric
       are identified using an index of type IppmMetricsRegistryIndex.

       Example:
       Given a multi-metrics measure of One-way-Delay(6) and One-way-
       Packet-Loss(12). The value of the field ippmNetMeasureMetrics is
       '0001000001000000'b, '1040'B. Results are logged in the
       ippmHistoryTable where One-way-Delay singletons have a value of
       ippmMetricsIndex of 6 while One-way-Packet-Loss singletons have a
       value of ippmMetricsIndex of 12.

       "
   ::= { ippmNetMeasureEntry 4 }

ippmNetMeasureBeginTime OBJECT-TYPE
   SYNTAX GMTTimeStamp
```

```
    MAX-ACCESS read-only
    STATUS     current
```

```
   DESCRIPTION
       "Specifies the time at which the measurement begins."
   ::= { ippmNetMeasureEntry 5 }


ippmNetMeasureCollectionRateUnit OBJECT-TYPE
   SYNTAX TimeUnit
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Specifies the unit of the measurement period."
   ::= { ippmNetMeasureEntry 6 }

ippmNetMeasureCollectionRate OBJECT-TYPE
   SYNTAX     Unsigned32
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
        "Gives the period used to collect singletons from the point of
        measures. This value is used as the cycle period in the report."
   ::= { ippmNetMeasureEntry 7 }


ippmNetMeasureDurationUnit OBJECT-TYPE
   SYNTAX TimeUnit
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Specifies the measurement duration unit."
   ::= { ippmNetMeasureEntry 8 }

ippmNetMeasureDuration OBJECT-TYPE
   SYNTAX     Unsigned32
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Specifies the measurement duration."
   ::= { ippmNetMeasureEntry 9 }

ippmNetMeasureHistorySize OBJECT-TYPE
   SYNTAX     Unsigned32
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Specifies the maximum number of results saved for each metric of
       this measure.
       Overflow condition will be managed by the object
       ippmNetMeasureResultsMgmt. "
```

```
   ::= { ippmNetMeasureEntry 10 }
```

```
ippmNetMeasureFailureMgmtMode OBJECT-TYPE
   SYNTAX     INTEGER {
       auto(1),
       manual(2),
       discarded(3)
   }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION
       "This object defines whether this row (and the measure controlled
       by this row) is restarted automatically, manually, or discarded
       upon failure, or reboot of the measurement system.
       'auto'
          The measure is restarted automatically.
       'manual'
          The measure has to be restarted manually.
       'discarded'
          The measure and it results are discarded.
       "
   ::= { ippmNetMeasureEntry 11 }

ippmNetMeasureResultsMgmt OBJECT-TYPE
   SYNTAX INTEGER {
       wrap(1),
       suspend(2)
   }
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "
       Action to take when the log is full. The measurement system owner
       may choose to either wrap, in which case the agent writes over
       existing records. The user may choose to suspend writing to the
       log in the event that he wishes to archive the data. The resume
       action causes the agent to begin to write in the log, and assumes
       the data has been cleared.
       This object indicates the way the measurement results are managed
       when the owner quota has been exceeded:
       'wrap'
          continue the measurement and erase the older entries in the
       history.
       'suspend'
          stop the measure and keep the results in the history.
       "
   ::= { ippmNetMeasureEntry 12 }

ippmNetMeasureSrcPacketType OBJECT-TYPE
   SYNTAX PacketType
```

```
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
```

```
        "Defines the type P of the source address of the packets sent by
        the measure."
    ::= { ippmNetMeasureEntry 13 }

ippmNetMeasureSrc OBJECT-TYPE
    SYNTAX PacketTypeAddress
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Specifies the address of the source of the measure.
        It is represented as a list of parameters corresponding to those
        of the PROTOCOL IDENTIFIER set in ippmNetMeasureSrcPacketType."
    ::= { ippmNetMeasureEntry 14}

ippmNetMeasureDstPacketType OBJECT-TYPE
    SYNTAX PacketType
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Defines the type P of the destination address of the packets
        sent by the measure."
    ::= { ippmNetMeasureEntry 15 }


ippmNetMeasureDst OBJECT-TYPE
    SYNTAX PacketTypeAddress
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "Specifies the address of the destination of the measure.
        It is represented as a list of parameters corresponding to those
        of the PROTOCOL IDENTIFIER set in ippmNetMeasureDstPacketType."
    ::= { ippmNetMeasureEntry 16 }

ippmNetMeasureTxMode OBJECT-TYPE
    SYNTAX INTEGER {
        other(0),
        periodic(1),
        poisson(2),
        multiburst(3)
           }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "The transmit mode used to send the packets:
        'other'
           The rule used to send the packets is unknown.
        'periodic'
```

Packets are sent periodically at ippmNetMeasureTxPacketRate
rate.
'poisson'

```
          Packets are sent using a Poisson law, the median is the value
       of ippmNetMeasureDevOrIntBurstSize, the deviation is
       ippmNetMeasureMedOrBurstSize.
       'multiburst'
          Packets are sent bursty at ippmNetMeasureTxPacketRate. The
       size of the burst is made of ippmNetMeasureMedOrBurstSize
       packets.
       Between 2 consecutive bursts, transmission stops during the time
       needed to send ippmNetMeasureDevOrIntBurstSize.

       "
   ::= { ippmNetMeasureEntry 17 }

ippmNetMeasureTxPacketRateUnit      OBJECT-TYPE
   SYNTAX TimeUnit
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The packet rate unit used to send the packets."
   ::= { ippmNetMeasureEntry 18 }

ippmNetMeasureTxPacketRate OBJECT-TYPE
   SYNTAX Unsigned32
   UNITS "Packets"
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The rate the packets are sent."
   ::= { ippmNetMeasureEntry 19 }

ippmNetMeasureMedOrBurstSize   OBJECT-TYPE
   SYNTAX     Unsigned32
   UNITS "Packets"
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "
       Multi-burst mode: This field represents the burst size in number
       of packets.
       Poisson mode: This field indicates the number of packets sent, on
       average, during each period corresponding to the median.
       The median is therefore
       MedOrBurstSize*TxPacketRateUnit/TxPacketRate.

       Example:
       If  TxPacketRateUnit/TxPacketRate is 100 packets/second, and if
       MedOrBurstSize, the number of packets sent during the period
       corresponding to the median is 50 packets, then the median equals
```

```
        50*1/100 = 1/2 seconds.
        "
    ::= { ippmNetMeasureEntry 20 }
```

```
ippmNetMeasureDevOrIntBurstSize   OBJECT-TYPE
   SYNTAX     Unsigned32
   UNITS "Packets"
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "
       Multi-burst mode: This field indicates the gap between 2 bursts,
       in number of packets.
       Example:
       If TxPacketRateUnit/TxPacketRate is 100 packets/second,
       and DevOrIntBurstSize equals 50 packets, then the gap between 2
       bursts is
       equal to 50*1/100, or 1/2 second.

       Poisson mode:
       This field indicates the typical difference between the packets
       of the period corresponding to the median.

       "
   ::= { ippmNetMeasureEntry 21 }



ippmNetMeasureLossTimeout OBJECT-TYPE
   SYNTAX     Unsigned32
   UNITS      "Milliseconds"
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
        "Specifies the delay after which the packet is considered lost
   by the sink."
   ::= { ippmNetMeasureEntry 22 }



ippmNetMeasureL3PacketSize OBJECT-TYPE
   SYNTAX     Unsigned32
   UNITS      "Bytes"
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Specifies the size of the packets counted at the IP network
       layer in regards to the PacketType definition.
       Example: For a PacketType 'ip ipip4' the L3 size will be the size
       of the packet at ipip4 level.
       "
   ::= { ippmNetMeasureEntry 23 }
```

```
ippmNetMeasureDataPattern OBJECT-TYPE
```

```
   SYNTAX     OCTET STRING
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The pattern used to fill the payload of the packet."

   ::= { ippmNetMeasureEntry 24 }


ippmNetMeasureTotalPktsRecv  OBJECT-TYPE
   SYNTAX Counter64
   UNITS "Packets"
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Reports the current number of packets received since the
       beginning of the measure.
       This parameters is useful to monitor the measure and it is needed
to compute statistics."
   ::= { ippmNetMeasureEntry 25 }


ippmNetMeasureLastUpdate OBJECT-TYPE
SYNTAX GMTTimeStamp
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The time when the last aggregation was computed."
   ::= { ippmNetMeasureEntry 26 }


ippmNetMeasureOperState OBJECT-TYPE
   SYNTAX INTEGER {
       unknown(0),
       running(1),
       stopped(2)
   }
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Reports the operational status of the network measure."
   ::= { ippmNetMeasureEntry 27 }

--
--
-- ippmAggrMeasure Group
--
--
```

--

```
--
-- ippmAggrMeasureTable
--
--

ippmAggrMeasureTable OBJECT-TYPE
   SYNTAX     SEQUENCE OF IppmAggrMeasureEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "An aggregated measure summarizes the results of previous network
       or aggregated measures. The results are saved in the
       ippmHistoryTable.
       Each step of the calculation for the measure produces a singleton
result per metric."
   ::= { ippmAggrMeasure 1 }

ippmAggrMeasureEntry OBJECT-TYPE
   SYNTAX     IppmAggrMeasureEntry
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "Typically, the configuration operation creates and sets the
       value of the fields of a new ippmAggrMeasureEntry.
       ippmAggrMeasureOwner and ippmAggrMeasureIndex identify the
       instance created.
       The field ippmAggrMeasureMetrics identifies the metric to
       compute. As such its ippmMetricsType should be 'aggregated'.

       The measure aggregates the results of a measure identified by
       ippmAggrMeasureHistoryOwner, ippmAggrMeasureHistoryIndex and
       ippmAggrMeasureHistoryMetric. The measure to aggregate belongs to
       ippmNetMeasureTable or ippmAggrMeasureTable.

       The aggregation starts at ippmAggrMeasureBeginTime and ends after
       ippmAggrMeasureDuration.

       An aggregated result is computed for each
       ippmMeasureCollectionRate tick and saved in the
       ippmHistoryTable."
   INDEX { ippmAggrMeasureOwner, ippmAggrMeasureIndex }
   ::= { ippmAggrMeasureTable 1 }


IppmAggrMeasureEntry ::= SEQUENCE {
   ippmAggrMeasureOwner                  IppmOwnerString,
   ippmAggrMeasureIndex                  IppmOwnerIndex,
   ippmAggrMeasureName                   SnmpAdminString,
```

```
   ippmAggrMeasureMetrics               IppmStandardMetrics,
   ippmAggrMeasureHistoryOwner          IppmOwnerString,
   ippmAggrMeasureHistoryIndex          IppmOwnerIndex,
```

```
   ippmAggrMeasureHistoryMetric          IppmMetricsRegistryIndex,
   ippmAggrMeasureFilter                 IppmMetricResultFilter,
   ippmAggrMeasureLowThreshold           Unsigned32,
   ippmAggrMeasureHighThreshold          Unsigned32,
   ippmAggrMeasureBeginTime              GMTTimeStamp,
   ippmAggrMeasureAggrPeriodUnit         TimeUnit,
   ippmAggrMeasureAggrPeriod             Unsigned32,
   ippmAggrMeasureDurationUnit           TimeUnit,
   ippmAggrMeasureDuration               Unsigned32,
   ippmAggrMeasureHistorySize            Unsigned32,
   ippmAggrMeasureStorageType            StorageType,
   ippmAggrMeasureAdminState             INTEGER,
   ippmAggrMeasureFastReport             OBJECT IDENTIFIER,
   ippmAggrMeasureResultsMgmt            INTEGER,
   ippmAggrMeasureLastUpdate             GMTTimeStamp,
   ippmAggrMeasureOperState              INTEGER,
   ippmAggrMeasureNbPktsTreated          Counter64,
   ippmAggrMeasureStatus                 RowStatus
}

ippmAggrMeasureOwner OBJECT-TYPE
   SYNTAX     IppmOwnerString
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "The owner who has configured this entry."
   ::= { ippmAggrMeasureEntry 1 }

ippmAggrMeasureIndex OBJECT-TYPE
   SYNTAX IppmOwnerIndex
   MAX-ACCESS not-accessible
   STATUS     current
   DESCRIPTION
       "The index of the aggregated measure. The value is managed by the
       owner."
   ::= { ippmAggrMeasureEntry 2 }

ippmAggrMeasureName OBJECT-TYPE
   SYNTAX SnmpAdminString
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "The name of the instance of the metric. It illustrates the
       specificity of the metric and includes the metric and the typeP.

       example: IP-port-HTTP-connectivity: free text."
   ::= { ippmAggrMeasureEntry 3 }
```

```
ippmAggrMeasureMetrics OBJECT-TYPE
   SYNTAX IppmStandardMetrics
```

```
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "ippmAggrMeasureMetrics defines the metrics to compute within
       this aggregated measure.

       Only aggregated metrics of the same type are allowed in this
       field (e.g. Measurement of minimum, average and maximum metrics
       are generally processed simultaneously on the same network
       measure).

       Results are saved in the ippmHistoryTable. Results of a metric
       are identified using an index of type IppmMetricsRegistryIndex.

       Example:
       Given a multi-aggregation of One-way-Delay-Median(9) and One-way-
       Delay-Minimum(10). The value of the field ippmAggrMeasureMetrics
       is '0000011000000000'b, '0600'B. Results are logged in the
       ippmHistoryTable where One-way-Delay-Median singletons have a
       value of ippmMetricsIndex of 9 while One-way-Delay-Minimum
       singletons have a value of ippmMetricsIndex of 10.

       NOTE WELL: It is not recommended to use the multi aggregation
       capability in conjunction with the filter feature.
   "
   ::= { ippmAggrMeasureEntry 4 }


ippmAggrMeasureHistoryOwner OBJECT-TYPE
   SYNTAX IppmOwnerString
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "The owner of the measure to summarize. "
   ::= { ippmAggrMeasureEntry 5 }

ippmAggrMeasureHistoryIndex OBJECT-TYPE
   SYNTAX IppmOwnerIndex
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "The owner index of the measure to summarize. "
   ::= { ippmAggrMeasureEntry 6 }

ippmAggrMeasureHistoryMetric OBJECT-TYPE
   SYNTAX IppmMetricsRegistryIndex
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
```

```
        "The metric of the measure to summarize. "
    ::= { ippmAggrMeasureEntry 7 }
```

```
ippmAggrMeasureFilter OBJECT-TYPE
   SYNTAX IppmMetricResultFilter
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "
       ippmAggrMeasureFilter defines the kind of filter to apply on a
       result to determine if the result is stored or not. The
       parameters of the filter are ippmAggrMeasureLowThreshold and
       ippmAggrMeasureHighThreshold.

       Thresholds have the same unit as the metric value.

       In the following examples we consider an aggregated measure. Its
       low threshold is set to 80.its high threshold is set to 100. The
       aggregation produced a flow of 12 aggregated results {40 30 60 85
       140 130 190 95 50 90 30 20}.

       If the filter is set to 'logInBandValue' then the results 85, 95,
       90 will be stored.

       If the filter is set to 'logOutBandValue' then the results 40 30
       60 140 130 190 50 30 20 will be stored.

       If the filter is set to 'logAboveValue' then the results 140 130
       190 will be stored.

       If the filter is set to 'logBelowValue' then the results 40 30 60
       50 30 20 will be stored.

       If the filter is set to 'logUpAndDownValue' then the results 40,
       140, 50 will be stored."
   ::= { ippmAggrMeasureEntry 8 }


ippmAggrMeasureLowThreshold OBJECT-TYPE
   SYNTAX Unsigned32
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "An event is generated when the result of the measure of the
       metric is lower that the value of ippmAggrMeasureLowThreshold.
       The threshold has the same unit as the metric. The metric unit is
       recorded in the object ippmMetricsUnit of this metric entry in
       the ippmMetricsTable.
       "
   ::= { ippmAggrMeasureEntry 9 }
```

```
ippmAggrMeasureHighThreshold OBJECT-TYPE
```

```
   SYNTAX Unsigned32
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "An event is generated when the result of the measure of the
       metric exceeds the value of ippmAggrMeasureHighThreshold.
       The threshold has the same unit as the metric. The metric unit is
       recorded in the object ippmMetricsUnit of this metric entry in
       the ippmMetricsTable.
       "
   ::= { ippmAggrMeasureEntry 10 }

ippmAggrMeasureBeginTime OBJECT-TYPE
   SYNTAX GMTTimeStamp
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "Specifies the time at which the aggregated measure starts."
   ::= { ippmAggrMeasureEntry 11 }


ippmAggrMeasureAggrPeriodUnit OBJECT-TYPE
   SYNTAX TimeUnit
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "Specifies the unit of the aggregated measure period."
   DEFVAL { second }
   ::= { ippmAggrMeasureEntry 12 }

ippmAggrMeasureAggrPeriod OBJECT-TYPE
   SYNTAX     Unsigned32
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "Specifies the amount of time between 2 measurement action
       intervals. The action is specific to the semantic of the measure.

       Network metrics:

       The ippmNetMeasureClockPattern transforms the flow of periodical
       instants as a flow of unpredictable instants of measurement
       packet emission.

       As the source and the sink share the definition of the clock of
       the measure, and as the sending timestamp is part of the
       measurement packet, the sink has the information to verify that
       the stream of packets generated by the source respects the clock
```

law.

Aggregated metrics:

```
       They are performed periodically on a sequence of results of other
       measures. The period corresponds to the interval between two
       successive computations of the metric. The value of
       ippmHistoryTimestamp result of a aggregated metric computed
       corresponds to the value of the ippmHistoryTimestamp of the last
       metric result of the sequence used to compute the aggregated
       metric."
   DEFVAL { 60 }

   ::= { ippmAggrMeasureEntry 13 }


ippmAggrMeasureDurationUnit OBJECT-TYPE
   SYNTAX TimeUnit
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "Specifies the unit of the measure duration."
   DEFVAL { second }
   ::= { ippmAggrMeasureEntry 14 }

ippmAggrMeasureDuration OBJECT-TYPE
   SYNTAX     Unsigned32
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "Specifies the duration of the measure."
   DEFVAL { 120 }
   ::= { ippmAggrMeasureEntry 15 }

ippmAggrMeasureHistorySize OBJECT-TYPE
   SYNTAX     Unsigned32
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "Specifies the maximum number of results saved for each metric of
       this measure. Overflow condition will be managed by the object
       ippmAggrMeasureResultsMgmt. "
   DEFVAL { 2 }
   ::= { ippmAggrMeasureEntry 16 }

ippmAggrMeasureStorageType OBJECT-TYPE
   SYNTAX     StorageType
   MAX-ACCESS  read-create
   STATUS      current
   DESCRIPTION
       "This object defines whether this row and the measure controlled
```

by this row are kept in volatile storage and lost upon reboot or
if this row is backed up
by non-volatile or permanent storage.

```
        Possible values are: other(1), volatile(2), nonVolatile(3),
        permanent(4), readOnly(5)."
    DEFVAL { nonVolatile }
    ::= { ippmAggrMeasureEntry 17 }

ippmAggrMeasureResultsMgmt OBJECT-TYPE
    SYNTAX INTEGER {
        wrap(1),
        suspend(2)
    }
    MAX-ACCESS read-only
    STATUS     current
    DESCRIPTION
        "This object displays the way the history of the aggregated
        measure is managed.
        'wrap'
           continue the measure and erase the older entries in the
        history.
        'suspend'
           stop the measure and keep the results in the history.
        "
    DEFVAL { wrap }
    ::= { ippmAggrMeasureEntry 18 }


ippmAggrMeasureAdminState OBJECT-TYPE
    SYNTAX INTEGER {
        start(0),
        stop(1)
    }
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
        "This object controls the activity of the aggregated measure.
        'start'
           The aggregated measure is started.
        'stop'
           The aggregated measure is stopped."
    DEFVAL { start }

    ::= { ippmAggrMeasureEntry 19 }


ippmAggrMeasureFastReport OBJECT-TYPE
    SYNTAX     OBJECT IDENTIFIER
    MAX-ACCESS read-create
    STATUS     current
    DESCRIPTION
```

"A fast report is required in order to verify quickly that a measure is running well.

```
       The feature 'fast report' is active if ippmAggrMeasureFastReport
       is not null and points to a notification.
       A fast report consists of sending by email to the owner of the
       measure, a table of the results of all the metrics computed by
       this aggregated measure. The owner email address is read from the
       ippmOwnersTable.

       ippmAggrMeasureFastReport identifies the notification which
       defines the header of the report.

       The results part of the report is made of a column of results per
       metrics. Results are separated using commas.

       To avoid disaster, an aggregated measure using a fast report must
       have a cycle of aggregation greater than or equal to 1 second and
       should not sent more than an email every 5 minutes and should not
       sent more than 12 emails."
   DEFVAL { zeroDotZero }
   ::= { ippmAggrMeasureEntry 20 }


ippmAggrMeasureLastUpdate OBJECT-TYPE
SYNTAX GMTTimeStamp
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "The time when the last aggregated measure was computed."
   ::= { ippmAggrMeasureEntry 21 }


ippmAggrMeasureOperState OBJECT-TYPE
   SYNTAX INTEGER {
       unknown(0),
       running(1),
       stopped(2)
   }
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
       "Reports the operational status of the aggregated measure."
   ::= { ippmAggrMeasureEntry 22 }


ippmAggrMeasureNbPktsTreated  OBJECT-TYPE
   SYNTAX Counter64
   UNITS  "Packets"
   MAX-ACCESS read-only
   STATUS     current
   DESCRIPTION
```

"Reports the current number of packets used to calculate the
aggregation since the start of the measure.

```
       This parameters is useful to monitor the measure and it is needed
       to compute statistics."
   ::= { ippmAggrMeasureEntry 23 }

ippmAggrMeasureStatus OBJECT-TYPE
   SYNTAX     RowStatus
   MAX-ACCESS read-create
   STATUS     current
   DESCRIPTION
       "The status of this entry. Once the entry status is set to
       active, the associate entry cannot be modified.
       "
   ::= { ippmAggrMeasureEntry 24 }



--
-- IPPM  Notifications
--


ippmAggrMeasureReport    NOTIFICATION-TYPE
   OBJECTS      {
       ippmAggrMeasureFilter,
       ippmAggrMeasureLowThreshold,
       ippmAggrMeasureHighThreshold,
       ippmMetricsType,
       ippmMetricsUnit,
       ippmMetricsDescription,
       ippmHistoryTimestamp,
       ippmHistoryValue,
       ippmHistoryPathToResults
   }
   STATUS       current
   DESCRIPTION
       "A notification sent because the value of the measure is under
       the high threshold value and greater than the low threshold
       value.
       The notification contains the instances of the ippmHistoryValue
       object that exceeded the threshold.
       The notification contains the instances of the
       ippmHistoryTimestamp identifying the time the event occurred.
       ippmHistoryPathToResults is a link to the file name, which
       contains detailed results corresponding to this event."
   ::= { ippmNotifications 1 }



ippmAggrMeasureHistoryFull    NOTIFICATION-TYPE
```

```
   OBJECTS      {
       ippmAggrMeasureName,
```

```
        ippmAggrMeasureHistorySize,
        ippmMetricsType,
        ippmMetricsUnit,
        ippmMetricsDescription,
        ippmHistoryTimestamp,
        ippmHistoryValue

    }
    STATUS       current
    DESCRIPTION
        "A notification sent when the size of the history of a metric of
        a aggregated measure exceeds ippmAggrMeasureHistorySize. The
        agent will then manage the reports according to the policy
        described in ippmAggrMeasureResultsMgmt."
    ::= { ippmNotifications 2 }

ippmNetMeasureHistoryFull    NOTIFICATION-TYPE
    OBJECTS {
        ippmNetMeasureName,
        ippmNetMeasureHistorySize,
        ippmMetricsType,
        ippmMetricsUnit,
        ippmMetricsDescription,
        ippmHistoryTimestamp,
        ippmHistoryValue
    }
    STATUS       current
    DESCRIPTION
        "A notification sent when the size of the history of a metric of
        a network measure exceeded ippmNetMeasureHistorySize. Then the
        agent manages the records according to the policy described in
        ippmNetMeasureResultsMgmt."
    ::= { ippmNotifications 3 }


--
-- IPPM MIB Conformance statements
--

ippmCompliances OBJECT IDENTIFIER ::={ ippmConformance 1 }

ippmGroups OBJECT IDENTIFIER ::={ ippmConformance 2 }

ippmProxyInterDomainCompliances         MODULE-COMPLIANCE
    STATUS             current
    DESCRIPTION
        "The compliance statement for SNMP entities which implement the
        IPPM MIB as a proxy in interdomain. The implementation of the
```

```
        VACM control is mandatory."
    MODULE -- this module
    MANDATORY-GROUPS {
```

```
        ippmSystemGroup, ippmHistoryGroup, ippmNetMeasureGroup,
        ippmAggrMeasureGroup, ippmNotificationGroup
    }
    ::= { ippmCompliances 1 }

ippmProxyCompliances         MODULE-COMPLIANCE
    STATUS             current
    DESCRIPTION
        "The compliance statement for SNMP entities which implement the
        IPPM MIB as a proxy."
    MODULE -- this module
    MANDATORY-GROUPS {
        ippmSystemGroup, ippmOwnersGroup, ippmHistoryGroup,
        ippmNetMeasureGroup, ippmAggrMeasureGroup, ippmNotificationGroup
    }
    GROUP ippmOwnersGroup
    DESCRIPTION
        "The ippmOwnersGroup is needed if VACM is not implemented."
    ::= { ippmCompliances 2 }


ippmEmbeddedCompliances  MODULE-COMPLIANCE
    STATUS             current
    DESCRIPTION
        "The compliance statement for SNMP entities which implement the
        IPPM MIB in a probe."
    MODULE -- this module
    MANDATORY-GROUPS {
        ippmSystemGroup, ippmHistoryGroup, ippmNetMeasureGroup
    }
    ::= { ippmCompliances 3 }


ippmSystemGroup    OBJECT-GROUP
    OBJECTS {
        ippmSystemSynchronizationDesc,
        ippmSystemTime,
        ippmSystemSynchronizationType,
        ippmSystemClockResolution,
        ippmSynchronizationTime,
        ippmSynchronizationStratum,
        ippmSynchronizationResolution,
        ippmPointOfMeasureMgmtAddrType,
        ippmPointOfMeasureMgmtAddress,
        ippmPointOfMeasureTestAddrType,
        ippmPointOfMeasureTestAddress,
        ippmSystemOperationalStatus,
        ippmSystemAggregatedMetrics,
```

```
        ippmPointOfMeasureMetrics,
        ippmMetricsType,
        ippmMetricsUnit,
```

```
        ippmMetricsDescription
    }
    STATUS  current
    DESCRIPTION
        "The IPPM System Group"
    ::=  { ippmGroups  1}


ippmNetMeasureGroup   OBJECT-GROUP
    OBJECTS  {
        ippmNetMeasureName,
        ippmNetMeasureMetrics,
        ippmNetMeasureBeginTime,
        ippmNetMeasureCollectionRateUnit,
        ippmNetMeasureCollectionRate,
        ippmNetMeasureDurationUnit,
        ippmNetMeasureDuration,
        ippmNetMeasureHistorySize,
        ippmNetMeasureFailureMgmtMode,
        ippmNetMeasureResultsMgmt,
        ippmNetMeasureSrcPacketType,
        ippmNetMeasureSrc,
        ippmNetMeasureDstPacketType,
        ippmNetMeasureDst,
        ippmNetMeasureTxMode,
        ippmNetMeasureTxPacketRateUnit,
        ippmNetMeasureTxPacketRate,
        ippmNetMeasureMedOrBurstSize,
        ippmNetMeasureDevOrIntBurstSize,
        ippmNetMeasureLossTimeout,
        ippmNetMeasureL3PacketSize,
        ippmNetMeasureDataPattern,
        ippmNetMeasureTotalPktsRecv,
        ippmNetMeasureLastUpdate,
        ippmNetMeasureOperState
    }
    STATUS  current
    DESCRIPTION
        "The IPPM Network Measure Group"
    ::=  { ippmGroups  2}


ippmHistoryGroup   OBJECT-GROUP
    OBJECTS  {
        ippmHistoryTimestamp,
        ippmHistoryValue,
        ippmHistoryPathToResults
    }
```

```
   STATUS  current
   DESCRIPTION
       "The IPPM History Group"
```

```
   ::=  { ippmGroups  3}

ippmAggrMeasureGroup       OBJECT-GROUP
   OBJECTS  {
       ippmAggrMeasureName,
       ippmAggrMeasureMetrics,
       ippmAggrMeasureBeginTime,
       ippmAggrMeasureAggrPeriodUnit,
       ippmAggrMeasureAggrPeriod,
       ippmAggrMeasureDurationUnit,
       ippmAggrMeasureDuration,
       ippmAggrMeasureFilter,
       ippmAggrMeasureLowThreshold,
       ippmAggrMeasureHighThreshold,
       ippmAggrMeasureHistorySize,
       ippmAggrMeasureStorageType,
       ippmAggrMeasureHistoryOwner,
       ippmAggrMeasureHistoryIndex,
       ippmAggrMeasureHistoryMetric,
       ippmAggrMeasureAdminState,
       ippmAggrMeasureFastReport,
       ippmAggrMeasureResultsMgmt,
       ippmAggrMeasureLastUpdate,
       ippmAggrMeasureOperState,
       ippmAggrMeasureNbPktsTreated,
       ippmAggrMeasureStatus
   }
   STATUS  current
   DESCRIPTION
       "The IPPM AggregatedMeasure Group"
   ::=  { ippmGroups 4}


ippmOwnersGroup    OBJECT-GROUP
   OBJECTS  {
       ippmOwnersGrantedMetrics,
       ippmOwnersQuota,
       ippmOwnersIpAddressType,
       ippmOwnersIpAddress,
       ippmOwnersEmail,
       ippmOwnersStatus
   }
   STATUS  current
   DESCRIPTION
       "The IPPM Owners Group"
   ::=  { ippmGroups 5}
```

```
ippmNotificationGroup       NOTIFICATION-GROUP
   NOTIFICATIONS  {
       ippmAggrMeasureReport,
```

```
        ippmNetMeasureHistoryFull,
        ippmAggrMeasureHistoryFull
    }
    STATUS  current
    DESCRIPTION
        "The IPPM Notification Group"
    ::=  { ippmGroups 6}

END
```

8 Security Considerations

8.1 VACM Access control

View Based Access Control, or VACM may be used to restrict access to certain objects, or even object instances within tables. For example, one may:

+ Give an 'administrator' write access to the ippmOwnersTable, whereas all other users may only have read access
+ Give access to individual rows in the network measure, aggregated measure, history, and report table to particular owners based upon indexing on an 'owners name', and even upon a particular measure. This will be illustrated below.
+ Give access of one ownerÆs measure, and associated results, to another owner in order to create an aggregated measure based upon the results.

8.1.1 Example of implementing VACM control for the IPPM-REPORTING-MIB

The following example illustrates how one could use VACM to restrict access to particular objects within the MIB. It uses syntax specific to a particular agent development toolkit, but may be generalized using the concepts as defined in the VACM MIB.

In this example, we have two NMS users, namely user1=owner1 and user2=owner2:

1) First we define the two users and their host addresses:

```
com2sec              owner1  owner1computer@ private
com2sec              owner2  owner2computer@ private
```

2) We then define SNMPv2c groups

```
group                        owner1  v2c     owner1
group                        owner2  v2c     owner2
view notif           included        ippmNotifications       ff
```

3.1) For the user owner1, we now define the views for which he will have read access

```
   # covers PointOfMeasureTable SynchronizationTable and all scalars
   view owner1read      included        ippmSystem              ff
```

```
   # covers OwnersTable
   view owner1read      included        ippmOwners              ff
   # covers MetricsTable
   view owner1read      included        ippmMeasure     ff
   # covers NetworkMeasureTable
   view owner1read      included
   ippmNetMeasureOwner.6.111.119.110.101.114.49 ff.df.c0
   # covers AggrMeasureTable
   view owner1read      included
        ippmAggrMeasureOwner.6.111.119.110.101.114.49   ff.df.c0

   3.2) We will now define the views for which owner1 will have write
   access
   view owner1write     included
        ippmAggrMeasureOwner.6.111.119.110.101.114.49   ff.df.c0
   # covers ReportSetupTable
   view owner1read      included
        ippmReportSetupOwner.6.111.119.110.101.114.49   ff.df.c0
   view owner1write     included
        ippmReportSetupOwner.6.111.119.110.101.114.49   ff.df.c0
   # covers HistoryTable
   view owner1read      included
        ippmHistoryMeasureOwner.6.111.119.110.101.114.49        ff.df.c0
   # covers ReportTable
   view owner1read      included
        ippmReportSequence.6.111.119.110.101.114.49     ff.df.c0

   3.3) For owner2, we will define the views for which he has read
   access
   view owner2read      included        ippmSystem              ff
   view owner2read      included        ippmOwners              ff
   view owner2read      included        ippmMeasure     ff
   # covers NetworkMeasureTable plus let's say the owner1 network
   measure of index X
   view owner2read      included
        ippmNetMeasureOwner.6.111.119.110.101.114.50    ff.df.c0
   view owner2read      included
        ippmNetMeasureOwner.6.111.119.110.101.114.49.X  ff.df.e0
   # covers AggrMeasureTable plus let's say the OWNER1 aggregated
   measure of index Y
   view owner2read      included
        ippmAggrMeasureOwner.6.111.119.110.101.114.50   ff.df.c0
   view owner2read      included
        ippmAggrMeasureOwner.6.111.119.110.101.114.49.Y ff.df.e0
   3.4) For owner2, we will define the views for which he has write
   access
   view owner2write     included
        ippmAggrMeasureOwner.6.111.119.110.101.114.50   ff.df.c0
```

```
   # covers ReportSetupTable
   view owner2read      included
        ippmReportSetupOwner.6.111.119.110.101.114.50   ff.df.c0
```

```
view owner2write     included
     ippmReportSetupOwner.6.111.119.110.101.114.50   ff.df.c0
# covers HistoryTable plus OWNER1 related X network measure results
and OWNER1 related Y aggregated measure results
view owner2read      included
     ippmHistoryMeasureOwner.6.111.119.110.101.114.50        ff.df.c0
view owner2read      included
     ippmHistoryMeasureOwner.6.111.119.110.101.114.49.X      ff.df.e0
view owner2read      included
     ippmHistoryMeasureOwner.6.111.119.110.101.114.49.Y      ff.df.e0
# covers ReportTable
view owner2read      included
     ippmReportSequence.6.111.119.110.101.114.50     ff.df.c0
```

3.5) Now we give the two users access to the views defined above. Note that owner1 and owner2 have read access to owner1read and owner2read views respectively. They have write access to owner1write and owner2write view respectively. And they both have access to all the notifications.

```
access       owner1  ""      any     noauth  exact   owner1read
     owner1write     notif
access       owner2  ""      any     noauth  exact   owner2read
     owner2write     notif
```

8.2 Privacy

The privacy concerns of network measurement are intrinsically limited by the active measurements. Unlike passive measurements, there can be no release of existing user data.

8.3 Measurement aspects

Conducting Internet measurements raises both security and privacy concerns. This memo does not specify an implementation of the metrics, so it does not directly affect the security of the Internet nor of applications that run on the Internet. However, implementations of these metrics must be mindful of security and privacy concerns.

There are two types of security concerns: potential harm caused by the measurements, and potential harm to the measurements. The measurements could cause harm because they are active, and inject packets into the network. The measurement parameters MUST be carefully selected so that the measurements inject trivial amounts of additional traffic into the networks they measure. If they inject "too much" traffic, they can skew the results of the measurement, and

in extreme cases cause congestion and denial of service.

The measurements themselves could be harmed by routers giving measurement traffic a different priority than "normal" traffic, or by an attacker injecting artificial measurement traffic. If routers can recognize measurement traffic and treat it separately, the measurements will not reflect actual user traffic. If an attacker injects artificial traffic that is accepted as legitimate, the loss rate will be artificially lowered. Therefore, the measurement methodologies SHOULD include appropriate techniques to reduce the probability measurement traffic can be distinguished from "normal" traffic.

Authentication techniques, such as digital signatures, may be used where appropriate to guard against injected traffic attacks.

8.4 Management aspects

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of read-write and/or read-only. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (for example by using IPSec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [18] and the View-based Access Control Model RFC 2575 [21] is recommended.

It is then a customer/user responsibility to ensure that the SNMP entity giving access to an instance of this MIB, is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

9 Document management

9.1 Open issues

Smilint complains when accessible-for-notify is used for an index.

9.2 Changes done since release 04

Report Group deleted:
reportHistoryTable deleted;
reportSetupTable deleted;
6 related notifications deleted;

low and high thresholds added in ippmAggrMeasureTable;

TC IppmOwnerIndex added to clearly define the owner namespace.

GMTTimestamp time origine changed to NTP (1900).

9.3 Changes done since release 03

+ SMI subtype: INTEGER vs Integer32...;
+ SMI UNITS: Clauses added;
+ cleanup of DEFVAL values;

+ Counter/index wrapping:
the index of the table wrap independently of the sequence of the results. That makes it very difficult for application to track the results. As the sequence id identify the instance of the result of a measure the index is removed both from the table and from the index clause.
ippmHistoryIndex removed from ippmHistoryEntry;
ippmHistoryIndex removed from the INDEX clause of the table ippmHistoryTable;
ippmReportIndex removed from ippmAggrHistoryEntry;
ippmReportIndex removed from the clause INDEX of ippmAggrHistoryEntry INDEX clause of the table ippmAggrHistoryTable;

9.4 Changes done since release 02

+ Security/VACM:
sharing table removed;
ippmMeasure merged with networkMeasure and AggrMeasure to have
all networkMeasure objects in read only.
Indexes belong to the table;
remove all reference to SNMPv1 ...inSNMPTrapPDU

+ System:
ippmSystemOperationalStatus added

ippmSynchronizationTable adapted for proxy mode:
ippmPointOfMeasureIndex added to the index of
ippmSystemCurrentSynchronization removed from system

capabilities:
ippmPointOfMeasureMetrics added to
IppmPointOfMeasureEntry;
ippmMetricsType added to ippmMetricsTable;

+ Owners
ippmMetricMaxHistorySize replaced with quota in ippmOwnersTable;

+ ippmOnHistoryFullAction replaced with resultsMgmt in aggr and network.;

+ network measure:
ippmNetMeasureOperState added to indicate the state of the network
measure
state;
added burst mode;
state of the measure: nb of singletons collected and oper status
added;

+aggregated metric:
fast report added to get raw results by email;

+ report setup:
onReportDeliveryClearHistory removed from
IppmMetricResultFilter;

+ Map field added to network, aggr and report tables to help to map
on topology map or admin view.

10 References

[1] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

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

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

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

[6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

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

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

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

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

[11] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.

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

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

[14] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol

(SNMPv2)", RFC 1905, January 1996.

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

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

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

11 Acknowledgments

A Kerbe.

12 Authors' Addresses

Emile STEPHAN
France Telecom R & D
2 avenue Pierre Marzin
F-22307 Lannion cedex
Phone: (+ 33) 2 96 05 11 11
Email: emile.stephan@francetelecom.com

Jessie Jewitt
France Telecom R & D
801 Gateway Blvd. Suit 500
South San Francisco, CA 94080
Tel: 1 650 875-1524
Email: jessie.jewitt@francetelecom.com

Full Copyright Statement

Internet organizations, except as needed for the purpose of
developing Internet standards in which case the procedures for