Network Working Group   E. Stephan/J. Jewitt
Internet Draft   France Telecom R&D
Document: draft-ietf-ippm-reporting-mib-02.txt   March 1st, 2003

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 measures 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 [6].

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 [7], STD 16, RFC 1212 [8] and RFC 1215 [9]. The second version, called SMIv2, is described in STD 58, RFC 2578 [10], STD 58, RFC 2579 [11] and STD 58, RFC 2580 [12].

Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [13]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [14] and RFC 1906 [15]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [15], RFC 2572 [16] and RFC 2574 [17].

Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [13]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [18].

A set of fundamental applications described in RFC 2573 [19] and the view-based access control mechanism described in RFC 2575 [20].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [21].

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. To address future needs specialized tables may be created, while augmenting the definition of the ippmMeasureTable.

The MIB architecture is inspired by the RMON model [xxiii],[xxiv] which specifies the MIB for the monitoring of a single point of measure. The IPPM-REPORTING-MIB differs from this model in that IPPM metrics measurement involves several points of measure and requires common references for time and for measure identification.

The IPPM-REPORTING-MIB introduces a framework where each application identifies its measures in an owner namespace. Using the namespace framework, an application may grant other owners access to its measurement results for aggregated metrics computation, reporting, or alarming.

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 to specify the relationship among the test protocol, the control protocol and 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. Moreover, it may be used to reduce the number of control frameworks.

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

Five types of data are introduced as a textual convention in this document: TypeP, TypePaddress, GMTTimeStamp, IppmStandardMetrics and IppmReportDefinition.

4.1.1. TypeP and TypePaddress

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. This naming convention serves as an important reminder that one must be conscious of the exact type of traffic being measured.

The standardization of the management of the IPPM measures relies on the capability to finely and 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 TypeP and the TypePaddress as new syntax in SMIv2 TEXTUAL-CONVENTION.

4.1.1.1. 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 TypeP 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 TypeP 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 TypeP 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".

4.1.2. 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.

4.1.3. IppmStandardMetrics

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

4.1.4. 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 consists of 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

The comparison of the measures results in a metric threshold that identifies particular measure values and times that directly impact service availability.

The comparison of the duration of repeated events with a duration threshold identifies particular measure values and times that directly affect an SLA.

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

A report is described using the TEXTUAL-CONVENTION IppmReportDefinition. 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 ippmReportSetupDefinition;
+ More elaborate reports are described using a metric threshold to generate alarms and events.
+ Pushing 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 IppmReportDefinition 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:

- ippmNotifications

- ippmOwnersGroup

- ippmSystemGroup

- ippmMeasureGroup

- ippmHistoryGroup

- ippmNetworkMeasureGroup

- ippmAggrMeasureGroup

- ippmReportGroup

4.2.1. The ippmOwners Group

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

4.2.2. The ippmSystem Group

This group consists of a set of parameters describing the clock synchronization at a particular point of measure over time.

This group is critical to the implementation of the IPPM MIB.

Section 6.3. of the IPPM Framework 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."

The aim of this group is to have these values available to compute reliable statistics. The implementation of this group is mandatory, whether the time synchronization is automatic or not.

4.2.3. The ippmMeasureGroup

This group displays all the measures configured on the measurement entity. It consists of the ippmMetricsTable and ippmMeasureTable. The ippmMeasureTable holds the common part of a measure, while the specific parameters are handled in the corresponding auxiliary table (ippmNetworkMeasure, ippmAggrMeasureTable...) .

The measurement entity describes in the ippmMetricsTable of the SNMP agent the local implementation of the standardized metrics. All standardized metrics should be displayed in this table, with the capability object defining whether the metric is implemented or not.

The control protocol registers a description of the existing measures in the ippmMeasureTable and in the auxiliary measure tables. The ippmMeasureTable table is read-create, but only allows for the creation of "aggregated" measures when defined in conjunction with the ippmAggrMeasureTable. Network measures are not allowed to be created directly by the management entity, and as such the measure table values for these measures should be display only.

The results of the measurements are logged in the ippmHistoryTable.

4.2.4. The ippmNetworkMeasure Group

The control protocol registers a description of the existing network measures in the ippmNetworkMeasureTable and in the ippmMeasureTable.

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

ippmNetworkMeasureTable is an auxiliary table of ippmMeasureTable, and is responsible for the configuration of the network measure.

4.2.5. The ippmAggrMeasure Group

ippmAggrMeasureTable is an auxiliary table of ippmMeasureTable, and is responsible for the consolidation of the results previously measured and saved in the ippmHistoryTable. The aggregated results are saved in the ippmHistoryTable and may be used for higher aggregated measures.

4.2.6. The Report Group

This group displays the existing reports of the measures collected.
ippmReportSetupTable is an auxiliary table of ippmMeasureTable, and is responsible for the configuration of the reports.
The reports are saved in the ippmReportTable, or sent directly to the applications.

4.2.7. The Notification Group

The Notification group specifies a list of valid notifications. They are used to push alarms or reports to the applications.

4.3. Row identification in an application namespace

The control protocol or the test protocol adds rows in the namespace of the corresponding measure.

An identifier of an instance of an object is defined as a list of objects in the clause INDEX. An object instance identifier in an owner namespace is defined as a list of objects in the clause INDEX where the first object type is IppmOwnerString.

As the OBJECT IDENTIFIER, which identifies the instance, begins with the owner value, the remaining values of the index fields may be chosen independently from one namespace to another.

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. This allows the owner to use the same values across MIB implementations.

Thus, it avoids polling to determine the next free index. Also, as a consequence, two applications will never find the same free index value.

The usage of owner namespace increases the speed of the management operations while reducing bandwidth consumption and CPU load in the agents and applications.

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.

As the MS manages its private range of indices, it simply chooses one when it wishes to create a new control entry. For the same reason, the setup of a measure on several points of measures consists of simply sending the same copy of the measure setup to the different points of measures involved.

4.4. Relationship of IPPM MIB tables

There is inherently a relationship between various tables in the IPPM 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 Measure Table

The owners table contains the list of "owners" that can create and activate remote IPPM measurements in an agent. As the table is "Read/Create", these users and their associated "access" rights on metric measurements can be directly configured. It is recommended to make use of "view based access control" in order to restrict access to this table. For example, the master user "acme" may be given "write" privileges on the ippmOwnersTable, whereas all others are restricted to "read" access. The user "acme" 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 measure table. Each entry in the measure table must be 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 the measure table.

```
+---------------------+              +---------------------------+
|    ippmOwnersTable  |              |    ippmMeasureTable       |
+---------------------+      1:N     +---------------------------+
| OwnersOwner: "Acme" |--------------| Measure Owner: "Acme"     |
|             "Foo"   |              | Measure Name:"OneWayDelay"|
|              .      |              |......                     |
|              .      |              | Measure Owner: "Foo"      |
|              .      |              | Measure Name: "PacketLoss"|
|                     |              | Measure Owner: "Foo"      |
+---------------------+              +---------------------------+
```

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

The network measure table and the aggregated measure table can be seen as logical "extensions" to the measure table.
The measure table contains information that is common to both types of measurements. The information found in the Network Measure Table and the Aggregated Measure Table is specific to each type of measure.

As the network measure table is read-only, 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. An entry can not be created in the network measure table without creating the corresponding entry in the measure table associated to the measure. This also implies that the "owner" of the measure be defined in the owners table.

The aggregated measure table allows for an "owner" to create aggregated measures (such as average, minimum, maximum) on existing measures that are in the measure table. If an "owner" (A) wishes to create an aggregated measure on a measure "owned" by another "owner" (B), then "owner" (B) must grant "owner" (A) access to his measures. This can be done in the resultsharing table.

Even though the Measure Table is read-create, an "owner" should only be able to create, or modify entries in the measure table that correspond to aggregated measure types. Should an "owner" attempt to update an entry in the measure table that corresponds to an entry in the network measure table, than access should be denied.

```
+---------------------------+    +----------------------------------+
|   ippmMeasureTable        |    |    ippmNetworkMeasureTable       |
+---------------------------+    +----------------------------------+
| Measure Owner: "Acme"     |    |  MeasureSrc: "Src1"              |
| Measure Name:"OneWayDelay | ---|  MeasureDst: "Dst1"              |
| .......                   |    |   ........                       |
| Measure Owner: "Foo"      |    |  MeasureSrc: "Src2"              |
| Measure Name: "PacketLoss"|    |  MeasureDst: "Dst2"              |
|                           |    +----------------------------------+
|                           |
|                           |    +----------------------------------+
|                           |    |   ippmAggrMeasureTable           |
|                           |    +----------------------------------+
| Measure Owner: "Acme"     |    |  AMHistoryOwner: "Foo"           |
| Measure Name: "AvgPLoss"  | ---|  AMHistoryMetric: "PacketLoss"   |
+---------------------------+    +----------------------------------+

+---------------------------+    +----------------------------------+
|  ippmHistoryTable         |    |   ippmResultSharingTable         |
|  (ex: with OWPL values)   |    |                                  |
+---------------------------+    +----------------------------------+
| Idx: Meas. Owner"Foo "    |    |  SharingOwner: "Foo"             |
|      Measure Index: 1     |    |  SharingMeasureOwner:"PacketLoss"|
|      Metrix Indx: 12      |    |                                  |
|                           |    |  SharingGrantedOwner:   "Acme"   |
|  HistorySqceNdx: 1        |    +----------------------------------+
|  GMTTimeStampValue        |
|  Value:       5           |
+---------------------------+
| Idx: Meas. Owner "Foo"    |
|      Mesure Index: 1      |
|      Metric Index: 12     |
|      HistorySqceNdx: 2    |
|   GMTTimeStampValue       |
|   Value:     15           |
| Idx: Meas. "Acme"         |
|      Measure Index: 3     |
|      Metric Index: 14     |
|     HistorySqceNdx: 1     |
|     GMTTimeStampValue     |
|  Value:        10         |
+---------------------------+
```

As the aggregated measure table essentially "inherits" from the measure table, one can not create an entry is this table without first creating an entry in the measure table. Likewise, one can not delete an entry in the measure table without first deleting the corresponding row in the aggregated measure table. This logic ensures that there are no "orphaned" table entries in the aggregated measure table.

5. IPPM-REPORTING-MIB conceptual presentation

5.1. IPPM-REPORTING-MIB diagram

Conceptual view of objects configured using the IPPM-REPORTING-MIB

```
        +--------------------------------------------------------+
        |                    IPPM-REPORTING-MIB entity           |
        |                                                        |
        |       +---------------------+ +-------------------+    |
        |       |                     | |                   |    |
        |       |  Measure scheduler  | |   Result storage  |    |
        |       |                     | |                   |    |
        |       |          ^          | | ^   ^^^           |    |
        |       |          |          | | |   |||           |    |
        |       +----------|----------+ +-|---|||-----------+    |
        |                  |              |   |||                |
        |       +----------|--------------|---|||-----------+    |
        |       |          |   owner      |   |||           |    |
        |       |          |   Acces      |   |||           |    |
        |       |          |  Control     |   |||           |    |
        |       +----------|--------------|---|||-----------+    |
        |                  |              |   |||                |
        +------------------|--------------|---|||----------------+
                           |              |   |||
                           |              |   |||
        +----------------+ |   +----------+-+ |||  +-------------+
        | ControlMeasure | |   | GetResult  | |||  | CreateResult|
        |----------------+-+   |------------| ||+--+-------------|
        |                |     |            | ||   |             |
        | owner          |     | owner      | ||   | owner       |
        | privateNdx     |     | privateNdx | ||   | privateNdx  |
        | metrics        |     | metric     | ||   | metrics     |
        | scheduler      |     | timestamp  | ||   | timestamp   |
        | addresses      |     +------------+ ||   | value       |
        | status         |                    ||   +-------------+
        +----------------+                    ||
                                              ||
                                              ||
                  +---------------------------+|
                  |                            |
        +---------+---------+           +------+-----------------+
        |GetMeasureResults  |           |GetMeasureMetricResults |
        |-------------------|           |------------------------|
        |                   |           |   owner                |
        | owner             |           |   privateNdx           |
        | privateNdx        |           |   metric               |
        +-------------------+           +------------------------+
```

The managed objects of the IPPM-REPORTING-MIB are the measures and the results.

5.2. Conceptual programming interface

This section describes a conceptual programming interface for the integration of the IPPM-REPORTING-MIB in a point of measure.

5.2.1. Measure control

A measure is created/deleted/suspended through the ControlMeasure() call.

5.2.2. Result log

A result of a measure is created in the IPPM-REPORTING-MIB History table using a CreateResult() call. Results belonging to a measure are managed according to the setup of the measure.

5.2.3. Reporting

Results are reported using the method GetResult(), GetMeasureMetricResults() and GetMeasureResults() respectively to get a singleton result, the singleton result of a metric measure, and finally to get the singleton result of a measure.

5.2.4. Logical calls

Objects are managed using 5 main primitives:

```
controlMeasure();
CreateResult();
GetResult();
GetMeasureMetricResults();
GetMeasureResults().
```

5.3. SNMP mapping

ControlMeasure() corresponds to a SNMP set-request on a conceptual row of ippmMeasureEntry and on a conceptual row of ippmNetworkMeasureEntry.

CreateResult() is a internal interface for adding measure results in the ippmHistoryTable.

GetResult() corresponds to an SNMP get-request on a result.

GetMeasureMetricResults() corresponds to a SNMP walk on the results of a metric measure subtree.

GetMeasureResults() corresponds to a SNMP walk on the results of a measure subtree.

6. Measurement architectures

There are four main measurement architectures.

6.1. Proxy architecture

```
         +----+                       +----+
         |NMS1|                       |NMS2|
         +----+                       +----+
           ^                           ^
           |                           |
           +----------+     +----------+
                      |     |
                  SNMP or Sibling
                      |     |
                      v     v
             +--------------------------+
             | IPPM-REPORTING-MIB agent |
             +--------------------------+
                      ^     ^
                      |     |
                    OWDP-Control
                      |     |
           +----------+     +----------+
           |                           |
           v                           v
+----------------+              +------------------+
| Packets-Sender |--OWDP-Test-->| Packets-Receiver |
+----------------+              +------------------+
```

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 accesses the results of its measurements in the IPPM-REPORTING-MIB statistics 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 either 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.

6.2. 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.

```
              +----+                               +----+
              |NMS1|                               |NMS2|
              +----+                               +----+
               ^  ^                                 ^  ^
               |  |                                 |  |
              SNMP|                                SNMP|
               |  |                                 |  |
               |  |                                 |  |
               | OWDP                               | OWDP
               |Control                             |Control
               |  |                                 |  |
               |  |     +------------------------------+
               |  |     |                           |  |
               |  |  +--|---------------------------+  |
               |  |  |  |                           |  |
               |  +--|--|------------------------+  |  |
               |  |  |  |                        |  |  |
               +--------+---------------------+  |  |
               |  |  |  |                     |  |  |  |
               |  |  |  |                     |  |  |  |
               v  v  v  v                     v  v  v  v
        +------------------+              +------------------+
        |IPPM-REPORTING-MIB|              |IPPM-REPORTING-MIB|
        |   agent          |              |     agent        |
        +------------------+              +------------------+
        |  Packets-Sender  |--OWDP-Test-->| Packets-Receiver |
        +------------------+              +------------------+
```

The activation of a measure by the control protocol or the test protocol creates a measure in the IPPM-REPORTING-MIB 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 on the fly 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 using a SNMP Trap PDU or a SNMP Inform PDU. The NMS may be the sender entity or the control entity;
+ 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 the result because it ensures that the entire block of the result is received. There is no control using SNMP Trap PDU.

6.3. Gateway architecture

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

```
             +-------+                                +------+
             | NMS1  |                                | NMS2 |
             +-------+                                +------+
               ^                                           ^
               |                                           |
             SNMP                                         SNMP
               |                                           |
               |  +----------------------------------------+
               |  |                                        |
               +-------------+          +------------------+
               |  |          |          |                  |
               +----------------------------------------+  |
               |  |          |          |               |  |
               |  |          v          v               |  |
               |  |     +------------------------+      |  |
               |  |     |  IPPM-REPORTING-MIB    |      |  |
               |  |     |     scheduler          |      |  |
               |  |     +------------------------+      |  |
               |  |     |    control server      |      |  |
               |  |     +------------------------+      |  |
               |  |          ^          ^               |  |
               |  |          |          |               |  |
               |  |      OWDP-Control protocol          |  |
               |  |          |          |               |  |
               |  |    +-----+          +-------+       |  |
               |  |    |                        |       |  |
               v  v    v                        v       v  v
        +-------------+---------+            +--------+-------------+
        |    IPPM-    | Packets |            |Packets |   IPPM      |
        |REPORTING-MIB| Sender  |            |Receiver|REPORTING-MIB|
        |  agent      |         |-OWDP-Test->|        |   agent     |
        +-------------+---------+            +--------+-------------+
```

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

6.4. 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 setups relies on the control and the test protocol security mechanisms.
- + The control of access to the results depends on the SNMP security mechanisms.

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

7. Reporting mode integration

The IPPM-REPORTING-MIB standardizes the parameters that:

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

7.1. Integration

The control protocol, test protocol and the IPPM-REPORTING-MIB share the same semantic.

The integration of the IPPM-REPORTING-MIB, and the test and control protocols, relies on the use of the conceptual programming interface described in section 6. It consists in pushing the measure setup/teardown parameters and the result values from the measurement software to the IPPM-REPORTING-MIB agent.

7.2. Setup of the measure

The creation of the measure consists only in transferring the measure description from the measurement software to the MIB. The management of the measure is done using the ControlMeasure().

The protocol, which provides the parameters of the measure to manage, may be the control protocol of the test protocol.

Different frameworks may be used to setup a measure.

7.2.1. Synchronous setup

The control protocol sets up the measure both in the sender and the receiver before the measurement.

7.2.2. Asynchronous setup

The control protocol sets up the measure only in the sender. In this case, the receiver has a service already activated (or pending)for the typeP of the measurement.

As the first test packet includes the description of the measure, it may differ from regular test packets. If the first test packet is not consistent with the regular test packets, it must not be used for performing metrics measurement.

7.3. Setup of the measurement report

The report description is an extension to the definition of a measure. It describes the event and the data to include in the report. A report is read by an NMS in the ippmReportTable, or pushed to a NMS using a SNMP Trap PDU, a SNMP Inform PDU, an email, or a SMS.

The control protocol, or the test protocol, includes the description of the report in the setup of the measure.

Different types of reports may be combined:

+ A trivial report defines the results to be saved in the ippmReportTable;
+ A basic report defines the host to which the results are pushed

on completion of the measure;
+ An alarm report defines a threshold on the results of the measure. A message is sent to a host when the result raises or

fall the threshold;
+ An SLA report defines a threshold on the results of the measure. The events are filtered using a staircase method. The

report consists in the results of the measure (time and value) of

the filtered events. The reports are sent at each measure

cycle
or when the measure completes.

7.4. Writing the results in the IPPM-REPORTING-MIB

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

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

Writing a result is done using the CreateResult().

7.5. Report download and upload

A report is read in the ippmReportTable using SNMP, or pushed by the IPPM_MIB agent using a SNMP Trap PDU, a SNMP Inform PDU, an email or a SMS.

7.6. Default value

The default values correspond to IP version 4.

8. Definition

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

IMPORTS
       MODULE-IDENTITY,
       NOTIFICATION-TYPE,
       OBJECT-TYPE,
       experimental ,Integer32
               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 "200203171200Z"    -- March 17, 2002
       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 the measures 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"

          ::= { experimental 10001 }

   ippm           OBJECT IDENTIFIER   ::= { experimental 10000 }

   --
   -- TEXTUAL-CONVENTION
   --

   IppmOwnerString ::= TEXTUAL-CONVENTION
          STATUS       current
          DESCRIPTION
                  "An OwnerString, which length is limited to 32."
          SYNTAX OCTET STRING (SIZE (0..32))


   TimeUnit ::= TEXTUAL-CONVENTION
          STATUS       current
          DESCRIPTION
                  "A list of time units."
          SYNTAX       INTEGER {
                  year(1),
                  month(2),
                  week(3),
                  day(4),
                  hour(5),
                  second(6),
                  millisecond(7),
                  microsecond(8),
                  nanosecond(9)
          }
   --
```

```
--

IppmStandardMetrics ::= TEXTUAL-CONVENTION
       STATUS      current
       DESCRIPTION
```

```
                  " Each standard metric is identified in the IPPM-METRICS-
                  REGISTRY under the node rfc in a chronological order. To
permit
                  several metrics to be performed in a single measure there is
an
                  need
                  to describe in a bit string the metrics to be performed,
                  granted...
                  This textual convention defines an octet string that gathered
in
                  a
                  bit string a sequence of bits. The bit order corresponds to
the
                  order
                  of the metrics 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



   GMTTimeStamp ::= TEXTUAL-CONVENTION
          STATUS       current
          DESCRIPTION
                  "The value of the ippmSystemTime object at which a specific
                  occurrence happened. The specific occurrence must be defined
in
                  the description of any object defined using this type.

                  field  octets  contents                  range
                  -----  ------  --------                  -----
                  1       1-4    second since 1 Jan 2000 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 directly inspired from the NTP timestamp format.
It differs because it counts the second since 1 Jan 2000 0H00 instead of 1 Jan 1900 0H00. The most significant bit of the part that represents the second is reserved. It will wrap in year 2068 (The NTP timestamp will wrap in year 2036).

```
                  This bit is set to indicate if the fractional part of the
second
                  contains a precision field and a synchronization field as
                  initially proposed in the OWAMP draft.

                  When this bit is not set the resolution is maximal.

                  The maximal resolution is close to 250 picoseconds.

                  The precision of the timestamp must be provided in another
field.
                  "
          SYNTAX       OCTET STRING (SIZE (8))


   TypeP  ::= 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.

                  TypeP is defined as a display string. It consists in 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 TypeP of the source address corresponding to telnet is
the
                  string 'ip.tcp.telnet'.

                  The TypeP of the source address corresponding to UDP packets
sent
                  in an IP tunnel is the string 'ip.ipip4.udp'.
                  Notes:
                  An IPPM measure is active, so generally a TypeP 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..255))

   TypePaddress ::= 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.

                  TypePaddress is defined as a display string. It consists in a
                  list of space separated parameter list. Each parameter in the
                  list corresponds a parameter of a PROTOCOL-IDENTIFIER of the
                  TypeP.
                  Example:
                  The TypeP 'ip.ipip4' has 2 parameters. A valid TypePaddress
value
                  is '192.168.1.1 128.2.6.7'."
          SYNTAX       OCTET STRING (SIZE (0..255))

   IppmReportDefinition ::= TEXTUAL-CONVENTION
          STATUS        current
          DESCRIPTION
          "IppmReportDefinition is intended to be used for describing the
          report resulting from a measurement. By default, all the results of a
          measure belong to the report of this measure.

          The first step of the report definition sets up triggers on the value
          of the measure, and on the distribution over time of the events
          generated by these triggers.

          The resulting measures corresponding to an event are reported
          periodically, or sent in alarms as soon as the event occurs.

          The end of the description describes housekeeping tasks.

          An action is performed if the corresponding bit is set to 1.

          onSingleton(1):
          The actions are performed each time a new result of the measure
          occurs.

          onMeasureCycle(2):
          The actions are performed on the results of the measure at the end of
          each cycle of measure.

          onMeasureCompletion(3):
          The actions are performed on the results of the measure at the end of
          the measure.

          reportOnlyUptoDownMetricResults(4):
          Report the contiguous results that are on opposite sides of the
          metric threshold.
```

reportOnlyExceededEventsDuration(5):
Report the current result of a series of contiguous results that exceed the metric threshold when the duration of the series is over the events duration threshold seconds.

```
       inIppmReportTable(6):
       Store the report in the local ippmReportTable.

       inSNMPTrapPDU(7):
       Send the report using a SNMP-Trap-PDU.

       inSNMPv2TrapPDU(8):
       Send the report using a SNMPv2-Trap-PDU.

       inInformRequestPDU(9):
       Send the report using a SNMP InformRequest-PDU.

       inEmail(10):
       Send the report using an email.

       inSMS(11):
       Send the report using a SMS.

       onReportDeliveryClearHistory(12):
       Remove all the results corresponding to this measure from the
       ippmHistoryTable when the report has been delivered.

       onReportDeliveryClearReport(13):
       Remove all the results corresponding to this measure from the
       ippmReportTable when the report has been delivered.
       "
       SYNTAX BITS {
               none(0), -- reserved
               onSingleton(1),
               onMeasureCycle(2),
               onMeasureCompletion(3),
               reportOnlyUptoDownMetricResults(4),
               reportOnlyExceededEventsDuration(5),
               inIppmReportTable(6),
               inSNMPTrapPDU(7),
               inSNMPv2TrapPDU(8),
               inInformRequestPDU(9),
               inEmail(10),
               inSMS(11),
               onReportDeliveryClearHistory (12),
               onReportDeliveryClearReport (13)
       }

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

```
--
-- IPPM  Conformance
```

```
--
ippmConformance      OBJECT IDENTIFIER   ::= { ippm 1 }

--
-- IPPM Mib objects definitions
--

ippmSystem           OBJECT IDENTIFIER   ::= { ippmReportingMib 1 }
ippmOwners           OBJECT IDENTIFIER   ::= { ippmReportingMib 2 }
ippmMeasure          OBJECT IDENTIFIER   ::= { ippmReportingMib 3 }
ippmHistory          OBJECT IDENTIFIER   ::= { ippmReportingMib 4 }
ippmNetworkMeasure   OBJECT IDENTIFIER   ::= { ippmReportingMib 5 }
ippmAggrMeasure      OBJECT IDENTIFIER   ::= { ippmReportingMib 6 }
ippmReport           OBJECT IDENTIFIER   ::= { ippmReportingMib 7 }


--
-- ippmSystem  Group
--
--


ippmSystemTime OBJECT-TYPE
       SYNTAX GMTTimeStamp
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
               "The current time of the measurement system."
       ::= { 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.

               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."
          ::= { ippmSystem 3 }

   ippmSystemClockResolution OBJECT-TYPE
          SYNTAX     Integer32
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "ippmSystemClockResolution provides the precision of the
clock
                  used for the measures. The unit is the picosecond. For
example,
                  the clock on an old Unix host might advance only once every
10
                  msec, and thus have a resolution of only 10 msec. So its
                  resolution is 100000 picosecond and the value of
                  ippmSystemClockResolution is 100000."
          ::= { ippmSystem 4 }

          ippmSystemCurrentSynchronization OBJECT-TYPE
          SYNTAX     Integer32
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "The index on the last synchronization event in the
                  ippmSynchronizationTable."
          ::= { ippmSystem 5 }


   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 point of measure. Each event is described in an
               ippmSynchronizationEntry.
               ippmSynchronizationTable is mandatory.
               ippmSynchronizationTable content is read only."
       ::= { ippmSystem 6 }

       ippmSynchronizationEntry OBJECT-TYPE
```

```
          SYNTAX     IppmSynchronizationEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "An entry describes a modification of the synchronization
status.
                  "
          INDEX { ippmSynchronizationIndex }
          ::= { ippmSynchronizationTable 1 }

   IppmSynchronizationEntry ::=
          SEQUENCE {
                  ippmSynchronizationIndex                Integer32,
                  ippmSynchronizationTime                 GMTTimeStamp,
                  ippmSynchronizationStratum              Integer32,
                  ippmSynchronizationResolution           Integer32
          }

   ippmSynchronizationIndex    OBJECT-TYPE
          SYNTAX     Integer32 (1 .. 65535)
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "An index that identifies the synchronization events in
                  chronological order."
          ::= { 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     Integer32
          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     Integer32
       UNITS      "NanoSeconds"
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
```

```
                  "The new time resolution computed after the synchronization
event
                  occured."
          ::= { ippmSynchronizationEntry 4 }

   ippmPointOfMeasureTable OBJECT-TYPE
          SYNTAX     SEQUENCE OF IppmPointOfMeasureEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  " A lookup table that identifies the management software in
                  charge of the point of measures.

                  ippmPointOfMeasureTable content is read only. It means that
the
                  measurement software handles the table internally

                  ippmPointOfMeasureTable is mandatory."
          ::= { ippmSystem 7 }

   ippmPointOfMeasureEntry OBJECT-TYPE
          SYNTAX     IppmPointOfMeasureEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  " An entry may be the management address of a 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. The
                  description may make the use of wildcards to define multiple
                  capabilities."
          INDEX { ippmPointOfMeasureIndex }
          ::= { ippmPointOfMeasureTable 1 }

   IppmPointOfMeasureEntry ::=
          SEQUENCE {
                  ippmPointOfMeasureIndex                Integer32,
                  ippmPointOfMeasureMgmtAddrType         InetAddressType,
                  ippmPointOfMeasureMgmtAddress          InetAddress,
                  ippmPointOfMeasureTypePAddress         TypeP,
                  ippmPointOfMeasureAddress              InetAddress
          }


   ippmPointOfMeasureIndex OBJECT-TYPE
          SYNTAX Integer32 (1 .. 65535)
          MAX-ACCESS not-accessible
```

```
       STATUS     current
       DESCRIPTION
               "The index of the entry."
       ::= { ippmPointOfMeasureEntry 1 }
```

```
   ippmPointOfMeasureMgmtAddrType OBJECT-TYPE
          SYNTAX InetAddressType
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "The type of address associated with 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 }


   ippmPointOfMeasureTypePAddress OBJECT-TYPE
          SYNTAX TypeP
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "Defines the type P of the address of the point of measure."
          DEFVAL { "ip" }
          ::= { ippmPointOfMeasureEntry 4 }

   ippmPointOfMeasureAddress OBJECT-TYPE
          SYNTAX InetAddress
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "Specifies the address of the point of measure.

                  It is represented as an octet string with specific semantics
and
                  length as identified by the ippmPointOfMeasureTypePAddress.

                  For example, if the ippmPointOfMeasureTypePAddress indicates
an
                  encapsulation of 'ip', this object length is 4, followed by
the 4
                  octets of the IP address, in network byte order."
          ::= { ippmPointOfMeasureEntry 5}

   --
   -- 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 create and activate remote
Ippm
                  measurements in an agent must previously be registered in the
                  ippmOwnersTable.
                  ippmOwnersTable content is read-create. It contains at least
the
                  owner 'monitor'. It is mandatory, except if the VACM
framework is
                  used."
          ::= { 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.

                  Notes:

                  The ippmOwnersIndex value is a local index managed directly
by
                  the agent. The management application must poll to get the
next
                  available index value.
                  It is not used in anyway in the other IPPM tables."
          INDEX { ippmOwnersIndex }
          ::= { ippmOwnersTable 1 }

   IppmOwnersEntry ::= SEQUENCE {
          ippmOwnersIndex              Integer32,
          ippmOwnersOwner              SnmpAdminString,
          ippmOwnersGrantedMetrics     IppmStandardMetrics,
          ippmOwnersGrantedRules       BITS,
          ippmOwnersIpAddressType      InetAddressType,
          ippmOwnersIpAddress          InetAddress,
          ippmOwnersEmail              SnmpAdminString,
          ippmOwnersSMS                SnmpAdminString,
          ippmOwnersStatus             RowStatus
   }
```

```
ippmOwnersIndex OBJECT-TYPE
       SYNTAX Integer32 (1.. 65535)
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
               "An arbitrary index that identifies an entry in this table"
       ::= { ippmOwnersEntry 1 }

ippmOwnersOwner OBJECT-TYPE
```

```
          SYNTAX     SnmpAdminString
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "The owner described by this entry."
          ::= { ippmOwnersEntry 2 }

   ippmOwnersGrantedMetrics OBJECT-TYPE
          SYNTAX     IppmStandardMetrics
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  " Defines the metrics granted to an owner."
          ::= { ippmOwnersEntry 3 }

   ippmOwnersGrantedRules OBJECT-TYPE
          SYNTAX     BITS {
                  all(0),
                  readonly(1),
                  permanent(2),
                  sender(3),
                  receiver(4),
                  report(5),
                  alarm(6)
          }
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "Defines the rules this owner may act on in the current IPPM
MIB
                  instance.
                  all(0):
                  The owner is granted all the rules.
                  readonly(1):
                  The measures (not only the metrics) that this owner may
access
                  are setup by the manager of the point of measure. The owner
can
                  not add new measures for these metrics. The creation and the
                  configuration of the measures corresponding to these metrics
are
                  managed by the manager of the point of measure.
                  permanent(2):
                  The measures (not only the metrics) that this owner may
access
                  are determined by the manager of the point of measure. The
owner
                  can not add new measures for these metrics. The creation and
```

the
first configuration of the measures corresponding to these
metrics are managed by the manager of the point of measure.
The
owner may modify the measures parameters of the entries of
the
corresponding ippmMeasureEntry whose access is read-write.
Typically this allows the owner to suspend the measures, to
change the beginning and end of the measures.

sender(3):

```
                  The owner may only activate measures for those metrics that
send
                  packets from the current point of measure. This flag is only
                  suitable for network measures. It shall be ignored for
derived
                  metrics.
                  receiver(4):
                  The owner may only activate measures for those metrics that
                  receive packets on the current point of measure. This flag is
                  only suitable for network measures. It shall be ignored for
                  derived metrics. Such control increases the security. The
owner
                  may not generate packets from the probe.

                  report(5):
                  The owner may setup aggregated metrics on the measures
                  corresponding to these metrics.

                  alarm(6):
                  The owner may setup alarms on the results of the measures
                  metrics.
                  e.g.:
                  if the owner Acme is granted with the metric Instantaneous-
                  Unidirectional-Connectivity as a Receiver in the current
point of
                  measure, then Acme can not setup a Instantaneous-
Unidirectional-
                  Connectivity to another point of measure."
          DEFVAL { 1 }
          ::= { ippmOwnersEntry 4 }

   ippmOwnersIpAddressType OBJECT-TYPE
          SYNTAX     InetAddressType
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "The IP address type of the management entity corresponding
to
                  this owner."
          ::= { ippmOwnersEntry 5 }

   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. The address is human readable and is represented using
the
                  dot format."
          ::= { ippmOwnersEntry 6 }

   ippmOwnersEmail OBJECT-TYPE
          SYNTAX     SnmpAdminString
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
```

```
                "The email address of the management entity corresponding to
this
                owner."
        ::= { ippmOwnersEntry 7 }

   ippmOwnersSMS OBJECT-TYPE
        SYNTAX     SnmpAdminString
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "The SMS phone number of the management entity corresponding
to
                this owner."
        ::= { ippmOwnersEntry 8 }

   ippmOwnersStatus OBJECT-TYPE
        SYNTAX     RowStatus
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "The status of this table entry."
        ::= { ippmOwnersEntry 9 }




   --
   --      ippmResultSharingTable
   --

   ippmResultSharingTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF IppmResultSharingEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                " The ippmResultSharingTable controls the access of an owner
to
                the measure results of other owners. An owner may grant
another
                access to read the result of its measure.

                Entries may exist in ippmResultSharingTable even if the
measures
                to be shared are not yet defined. Deleting a measure entry in
the
                ippmMeasureTable does not delete the entries corresponding to
                this measure in the ippmResultSharingTable. This table is
                optional.
```

```
                  ippmResultSharingTable content is read-create.

                  If this table is not implemented then the owner has only
access
                  to its own measurement results."
          ::= { ippmOwners 2 }

   ippmResultSharingEntry OBJECT-TYPE
```

```
       SYNTAX     IppmResultSharingEntry
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
               "An entry allows an owner to read the results of a measure
owned
               by another owner.
               It permits 2 typical usages:
               1) Creating derived measurements on these results
               2) Reading the results from a remote management station.

               Example: if acme.12 is a One-way-Delay(6) measure, Acme may
allow
               Peter to make derived metrics on the results of this
measure."
       INDEX { ippmResultSharingOwner, ippmResultSharingIndex}
       ::= { ippmResultSharingTable 1 }


   IppmResultSharingEntry ::= SEQUENCE {
       ippmResultSharingOwner                IppmOwnerString,
       ippmResultSharingIndex                Integer32,
       ippmResultSharingMeasureOwner         IppmOwnerString,
       ippmResultSharingMeasureIndex         Integer32,
       ippmResultSharingGrantedOwner         IppmOwnerString,
       ippmResultSharingStatus               RowStatus
   }
   ippmResultSharingOwner OBJECT-TYPE
       SYNTAX IppmOwnerString
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
               " The owner of this result control entry. Typically the owner
who
               created this conceptual row."
       ::= { ippmResultSharingEntry 1 }


   ippmResultSharingIndex OBJECT-TYPE
       SYNTAX Integer32 (1.. 65535)
       MAX-ACCESS not-accessible
       STATUS     current
       DESCRIPTION
               " The index of this result control entry. The value is
managed by
               the owner. On creation a SNMP error 'inconsistentValue' is
               returned if this value is already in use by this owner."
       ::= { ippmResultSharingEntry 2 }
```

```
ippmResultSharingMeasureOwner OBJECT-TYPE
       SYNTAX IppmOwnerString
       MAX-ACCESS read-create
       STATUS     current
       DESCRIPTION
```

```
                "The owner of the measure to be shared. The couple
                ippmResultSharingMeasureOwner, ippmResultSharingMeasureIndex
                identifies absolutely a measure"
        ::= { ippmResultSharingEntry 3 }

ippmResultSharingMeasureIndex OBJECT-TYPE
        SYNTAX Integer32 (1.. 65535)
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "The index of the measure to be shared."
        ::= { ippmResultSharingEntry 4 }

ippmResultSharingGrantedOwner OBJECT-TYPE
        SYNTAX IppmOwnerString
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION

                "The owner who is granted access to the result of the measure
                described by the couple ippmResultSharingMeasureOwner,
                ippmResultSharingMeasureIndex."
        ::= { ippmResultSharingEntry 5 }

ippmResultSharingStatus OBJECT-TYPE
        SYNTAX RowStatus
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                " The status of this table entry. Once the entry status is
set to
                active."
        ::= { ippmResultSharingEntry 6 }


--

--
--
-- ippmMeasure  Group
--
--
--

ippmMetricTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF IppmMetricEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
```

"This table describes the current implementation and is mandatory. Each IPPM standardized metric must be described in the table.

```
                In reporting mode, the entries of this table may be not
                accessible. It means that the measurement software handles
the
                table internally.

                ippmMetricTable is mandatory.
                ippmMetricTable content is read only."
        ::= { ippmMeasure 1 }

   ippmMetricEntry OBJECT-TYPE
        SYNTAX     IppmMetricEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "An entry describes the static capabilities of a metric
                implementation."
        INDEX { ippmMetricIndex }
        ::= { ippmMetricTable 1 }

   IppmMetricEntry ::=
   SEQUENCE {
        ippmMetricIndex           Integer32,
        ippmMetricCapabilities    INTEGER,
        ippmMetricUnit            INTEGER,
        ippmMetricDescription     SnmpAdminString,
        ippmMetricMaxHistorySize  Integer32
   }

   ippmMetricIndex OBJECT-TYPE
        SYNTAX Integer32 (1.. 65535)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "ippmMetricIndex defines an unambiguous index for each
                standardized metric. Its value is the value of the node of
the
                metric in the IPPM-REPORTING-MIB metrics registry
                ippmMib.metrics.rfc.
                Each metric registered in the standard registry must be
present
                in this table.
                This index is used to identify the metric calculated between
the
                IPPM-REPORTING-MIB entities involved in the measure.
                Example:
                The index of the metric onewayPacketLossAverage which is
                registered as ippmMib.metrics.rfc.onewayPacketLossAverage
will
```

```
                always have the value 14."
        ::= { ippmMetricEntry 1 }


ippmMetricCapabilities OBJECT-TYPE
        SYNTAX INTEGER {
        notImplemented(0),
        implemented(1)
```

```
          }
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "A value of notImplemented implies the metric is not
implemented.
                  A value of implemented implies the metric is implemented."
          DEFVAL { implemented }
          ::= { ippmMetricEntry 2 }

   ippmMetricUnit OBJECT-TYPE
          SYNTAX INTEGER {
                  noUnit(0),
                  second(1),
                  ms(2),
                  us(3),
                  ns(4),
                  percentage(5),
                  packets(6),
                  byte(7),
                  kbyte(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."
          ::= { ippmMetricEntry 3 }

   ippmMetricDescription OBJECT-TYPE
          SYNTAX SnmpAdminString
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "A textual description of the metric implementation."
          ::= { ippmMetricEntry 4 }



          ippmMetricMaxHistorySize OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-only
          STATUS current
          DESCRIPTION
                  "Specifies the maximum number of results that a metric
measure
                  can save in the ippmHistoryTable."
```

```
        DEFVAL { 200 }
        ::= { ippmMetricEntry 5 }
```

```
   --
   -- ippmMeasureTable
   --
   --

   ippmMeasureTable OBJECT-TYPE
          SYNTAX     SEQUENCE OF IppmMeasureEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "The table of all the IPPM measures which are running in the
                  device. They may not all be active.

                  A measure consists of a subset of metrics to compute. The
results
                  of the measure may be saved in the ippmHistoryTable. The
                  configuration of the measure sets the size of the history
                  requested in ippmMeasureHistorySize.

                  The maximum number of MIB objects to be collected in the
portion
                  of ippmHistoryTable associated with this metric depends on
the
                  value of the ippmMetricMaxHistorySize.

                  The value of each metric ippmMeasureHistorySize must not be
over
                  the value of ippmMetricMaxHistorySize corresponding to this
                  metric in the ippmMetricTable.

                  The ippmMeasureTable is mandatory.

                  ippmMeasureTable content is read-create. The table is handled
                  internally by the measurement software for network measures.

                  The setup of network is not permitted through the IPPM
REPORTING
                  MIB. OWAP provides a setup protocol to enable and teardown
                  networks measures."
          ::= { ippmMeasure 2 }

   ippmMeasureEntry OBJECT-TYPE
          SYNTAX     IppmMeasureEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "The measure entries are created/deleted internally by the
                  measurement software."
          INDEX { ippmMeasureOwner, ippmMeasureIndex }
```

```
       ::= { ippmMeasureTable 1 }

IppmMeasureEntry ::=
       SEQUENCE {
               ippmMeasureOwner             IppmOwnerString,
               ippmMeasureIndex             Integer32,
               ippmMeasureName              SnmpAdminString,
                ippmMeasureMetrics           IppmStandardMetrics,
                ippmMeasureBeginTime         GMTTimeStamp,
                ippmMeasureClockPeriodUnit   TimeUnit,
                ippmMeasureClockPeriod       Integer32,
                ippmMeasureDurationUnit      TimeUnit,
                ippmMeasureDuration          Integer32,
                ippmMeasureHistorySize       Integer32,
                ippmMeasureStorageType       StorageType,
                ippmMeasureStatus            RowStatus
        }


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

   ippmMeasureIndex OBJECT-TYPE
        SYNTAX Integer32 (1.. 65535)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "The owner index of the measure. The value is managed by the
                owner."
        ::= { ippmMeasureEntry 2 }

   ippmMeasureName 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"
        ::= { ippmMeasureEntry 3 }



   ippmMeasureMetrics OBJECT-TYPE
        SYNTAX IppmStandardMetrics
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "Defines the metrics to compute within this measure. A
```

```
measure
                  may be configured for the result of different metric
singletons
                  to be archived in the ippmHistoryTable. The ippmMetricIndex
of
                  the created result has the value of the bit index of the
                  corresponding ippmMeasureMetrics as explained above in the
                  ippmMetricIndex definition.

                  Example:
                  A measure asking for One-way-Delay(6) and One-way-Packet-
Loss(12)
                  generated a flow of singletons which are logged in the
                  ippmHistoryTable. The singletons created for the One-way-
Delay
                  measure have a value of ippmMetricIndex of 6 while the
created
                  singletons for the One-way-Packet-Loss measure have a value
of
                  ippmMetricIndex of 12."
          -- { one-way-Delay, one-way-Packet-Loss }
          DEFVAL { '0001000001000000'b }
          ::= { ippmMeasureEntry 4 }

   ippmMeasureBeginTime OBJECT-TYPE
          SYNTAX GMTTimeStamp
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "Specifies the time at which the measure starts."
          ::= { ippmMeasureEntry 5 }


   ippmMeasureClockPeriodUnit OBJECT-TYPE
          SYNTAX TimeUnit
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "Specifies the unit of the measure period."
          DEFVAL { second }
          ::= { ippmMeasureEntry 6 }

   ippmMeasureClockPeriod OBJECT-TYPE
          SYNTAX     Integer32
          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 ippmNetworkMeasureClockPattern 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, as the sending timestamp is part of the measurement packet, the sink have 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 in to compute the
aggregated
                  metric."
          DEFVAL { 60 }
          ::= { ippmMeasureEntry 7 }






   ippmMeasureDurationUnit OBJECT-TYPE
          SYNTAX TimeUnit
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "Specifies the unit of the measure duration."
          DEFVAL { second }
          ::= { ippmMeasureEntry 8 }

   ippmMeasureDuration OBJECT-TYPE
          SYNTAX     Integer32
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "Specifies the duration of the measure."
          DEFVAL { 120 }
          ::= { ippmMeasureEntry 9 }

   ippmMeasureHistorySize OBJECT-TYPE
          SYNTAX     Integer32
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "Specifies the maximum number of results saved for each
metric of
                  this measure. The history of each metric is managed as a
circular
                  table. The newest result overwrites the oldest one when the
                  history granted to this metric measure is full.
                  The management of the results may be optimized if
synchronized
                  with the reports steps of this measure. "
          DEFVAL { 120 }
          ::= { ippmMeasureEntry 10 }

   ippmMeasureStorageType 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 }
        ::= { ippmMeasureEntry 11 }


   ippmMeasureStatus OBJECT-TYPE
        SYNTAX     RowStatus
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "The status of this table entry. Once the entry status is set
to
                active, the associate entry cannot be modified."
        ::= { ippmMeasureEntry 12 }

   --
   -- ippmHistory  Group
   --
   --

   --
   -- ippmHistoryTable
   --


   ippmHistoryTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF IppmHistoryEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "The table of the results of the measures."
        ::= { 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 ippmMeasureOwner, ippmMeasureIndex,
ippmMetricIndex
                  and ippmHistoryIndex.
```

```
                In the index :

                + ippmMeasureOwner identifies the owner of the measure;

                + ippmMeasureIndex identifies the measure in the owner
namespace;

                + ippmMetricIndex identifies the metric measured in
                ippmMetricTable;

                + ippmHistoryIndex is the local index of the result on the
                history table."
        INDEX { ippmMeasureOwner, ippmMeasureIndex, ippmMetricIndex,
        ippmHistoryIndex }
        ::= { ippmHistoryTable 1 }

   IppmHistoryEntry ::=
        SEQUENCE {
                ippmHistoryIndex             Integer32,
                ippmHistorySequence          Integer32,
                ippmHistoryTimestamp         GMTTimeStamp,
                ippmHistoryValue             Integer32
        }

   ippmHistoryIndex OBJECT-TYPE
        SYNTAX Integer32 (1.. 65535)
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                " A local index that only identifies a result in the history
                table."
                ::= { ippmHistoryEntry 1 }

                ippmHistorySequence OBJECT-TYPE
                SYNTAX Integer32 (1.. 65535)
                MAX-ACCESS read-only
                STATUS     current
                DESCRIPTION
                "ippmHistorySequence is the sequence index of the measurement
                results of the measure of a metric.

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

                Aggregated metrics:
```

```
              It is the sequence index of the aggregated metric results
              computed."
       ::= { ippmHistoryEntry 2 }
```

```
   ippmHistoryTimestamp OBJECT-TYPE
          SYNTAX GMTTimeStamp
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "The instant of the measure of the result."
          ::= { ippmHistoryEntry 3 }

   ippmHistoryValue OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "The value of the measure."
          ::= { ippmHistoryEntry 4 }

   ippmOnHistoryFullAction   OBJECT-TYPE
          SYNTAX  INTEGER {
          wrap(1),
          suspend(2),
          resume(3)
          }
          MAX-ACCESS read-write
          STATUS     current
          DESCRIPTION
                  "Action to take when the history log is full. The user 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 history log, and
                  assumes the data has been cleared."
          ::= { ippmHistory 2 }




   --
   -- ippmNetworkMeasure   Group
   --


   --
   --
   -- ippmNetworkMeasureTable
   --
   --
```

```
ippmNetworkMeasureTable OBJECT-TYPE
       SYNTAX     SEQUENCE OF IppmNetworkMeasureEntry
       MAX-ACCESS not-accessible
       STATUS     current
```

```
          DESCRIPTION
                  "A entry is a measure which performs network measures and
                  provides a flow of results.

                  This table extends the ippmMeasureTable.

                  It performs several metric measurements per packet exchange.
Each
                  step of a measure produces a singleton result per metric. The
                  time of the measure and the value of the metric are saved in
the
                  ippmHistoryTable."
          ::= { ippmNetworkMeasure 1 }

   ippmNetworkMeasureEntry OBJECT-TYPE
          SYNTAX     IppmNetworkMeasureEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  " Typically the configuration operation sets both the values
of
                  the new ippmMeasureEntry and of the new
IppmNetworkMeasureEntry.

                  IppmNetworkMeasureTable is mandatory.

                  IppmNetworkMeasureTable content is read only. It means that
the
                  measurement software handles the table internally. The setup
of
                  network is not permitted through the IPPM REPORTING MIB. OWAP
                  provides a setup protocol to enable and teardown networks
                  measures.

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

                  The ippmNetworkMeasureTable typical usage consists is
providing
                  network measure indexes to permits aggregated measure to
perform
                  aggregation on the results of network measures.
                  An obvious usage of the ippmNetworkMeasureTable consists in
the
```

```
                verification of the network measures states."
        INDEX { ippmMeasureOwner, ippmMeasureIndex }
        ::= { ippmNetworkMeasureTable 1 }

IppmNetworkMeasureEntry ::=
        SEQUENCE {
                ippmNetworkMeasureSrcTypeP            TypeP,
                ippmNetworkMeasureSrc                 TypePaddress,
                ippmNetworkMeasureDstTypeP            TypeP,
                ippmNetworkMeasureDst                 TypePaddress,
                ippmNetworkMeasureClockPattern        OCTET STRING,
                ippmNetworkMeasurePoissonRate         Integer32,
                ippmNetworkMeasureTimeoutDelay        Integer32,
                ippmNetworkMeasureL3PacketSize        Integer32,
                ippmNetworkMeasureDataPattern         OCTET STRING
        }



   ippmNetworkMeasureSrcTypeP OBJECT-TYPE
        SYNTAX TypeP
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "Defines the type P of the source address of the packets sent
by
                the measure."
        DEFVAL { '04000080001000'H } -- ->ip: 4.0.0.8.0.1.0
        ::= { ippmNetworkMeasureEntry 1 }

   ippmNetworkMeasureSrc OBJECT-TYPE
        SYNTAX TypePaddress
        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 sets in
ippmNetworkMeasureSrcTypeP."
        ::= { ippmNetworkMeasureEntry 2}

   ippmNetworkMeasureDstTypeP OBJECT-TYPE
        SYNTAX TypeP
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "Defines the type P of the destination address of the packets
                sent by the measure."
        ::= { ippmNetworkMeasureEntry 3 }


   ippmNetworkMeasureDst OBJECT-TYPE
        SYNTAX TypePaddress
        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 sets in
ippmNetworkMeasureSrcTypeP."
```

```
       ::= { ippmNetworkMeasureEntry 4 }

ippmNetworkMeasureClockPattern OBJECT-TYPE
       SYNTAX OCTET STRING
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
```

```
                  "This cyclic clock shapes the profile of the instants of
                  measurement action provided by ippmMeasureClockPeriod
according
                  to an arbitrary distribution law. The clock resolution is
                  ippmMeasureClockPeriod. The bits of the clock pattern set to
the
                  value 1 determine the valid instants of measurement action. A
                  measure is to be processed if and only if the current bit
value
                  is 1.
                  This pseudo-random clock pattern allows the configuration by
the
                  NMS of numerous kind of time sampling law such as periodic,
                  pseudo random or Poisson.

                  The source of the measure sends the stream of measurement
packets
                  synchronously with the stream of instants selected by the
clock
                  pattern sampling.

                  ippmNetworkMeasureClockPattern can not be used conjointly
with
                  ippmNetworkMeasurePoissonRate."
          DEFVAL { "11111111" }
          -- 100% periodic
          ::= { ippmNetworkMeasureEntry 5 }

   ippmNetworkMeasurePoissonRate   OBJECT-TYPE
          SYNTAX     Integer32
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "Indicates the average number of packets per seconds sent
using a
                  poisson law.

                  ippmNetworkMeasurePoissonRate can not be used conjointly with
                  ippmNetworkMeasureClockPattern."
          DEFVAL { 30 }
          ::= { ippmNetworkMeasureEntry 6 }

   ippmNetworkMeasureTimeoutDelay OBJECT-TYPE
          SYNTAX     Integer32
          MAX-ACCESS read-only
          STATUS     current
          -- UNITS     "Milliseconds"
          DESCRIPTION
```

```
       "Specifies the delay after which the packet is considered lost by the
       sink."
       DEFVAL { 1 }
       ::= { ippmNetworkMeasureEntry 7 }

ippmNetworkMeasureL3PacketSize OBJECT-TYPE
       SYNTAX     Integer32
       MAX-ACCESS read-only
       STATUS     current
       DESCRIPTION
```

```
                 "Specifies the size of the packets sent at the last network
layer
                 in regards to the TypeP definition."
         DEFVAL { 64 }
         ::= { ippmNetworkMeasureEntry 8 }

   ippmNetworkMeasureDataPattern OBJECT-TYPE
         SYNTAX     OCTET STRING
         MAX-ACCESS read-only
         STATUS     current
         DESCRIPTION
                 "The current field defines the round robin pattern used to
fill
                 the packet."
         DEFVAL { 'FF'H }
         ::= { ippmNetworkMeasureEntry 9 }




   --
   --
   -- ippmAggrMeasure   Group
   --
   --
   --
   --
   -- ippmAggrMeasureTable
   --
   --

   ippmAggrMeasureTable OBJECT-TYPE
         SYNTAX     SEQUENCE OF IppmAggrMeasureEntry
         MAX-ACCESS not-accessible
         STATUS     current
         DESCRIPTION
                 " This table extends the ippmMeasureTable.
                 An aggregated measure summarizes the results of previous
network
                 or aggregated measures. The results may be 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 sets both the values
of

                  the new ippmMeasureEntry and of the new IppmAggrMeasureEntry.

                  ippmAggrMeasureTable is mandatory.
```

```
                ippmAggrMeasureTable content is read only. It means that the
                measure software handles the table internally.

                The ippmMeasureMetrics defines the metric to compute.
                The results of the measure to summarize are identified by:
                + ippmAggrMeasureHistoryOwner,
                + ippmAggrMeasureHistoryOwnerIndex and
                + ippmAggrMeasureHistoryMetric

                The aggregated task starts at ippmMeasureBeginTime and ends
after
                ippmMeasureDuration. An aggregated result is performed and
saved
                in the ippmHistoryTable for each ippmMeasureClockPeriod tick.
"
        INDEX { ippmMeasureOwner, ippmMeasureIndex }
        ::= { ippmAggrMeasureTable 1 }


   IppmAggrMeasureEntry ::=
        SEQUENCE {
                ippmAggrMeasureHistoryOwner           IppmOwnerString,
                ippmAggrMeasureHistoryOwnerIndex      Integer32,
                ippmAggrMeasureHistoryMetric          Integer32
        }


   ippmAggrMeasureHistoryOwner OBJECT-TYPE
        SYNTAX IppmOwnerString
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "The owner of the measure to summarize. "
        ::= { ippmAggrMeasureEntry 1 }

   ippmAggrMeasureHistoryOwnerIndex OBJECT-TYPE
        SYNTAX Integer32 (1.. 65535)
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "The owner index of the measure to summarize. "
        ::= { ippmAggrMeasureEntry 2 }

   ippmAggrMeasureHistoryMetric OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "The metric of the measure to summarize. "
```

```
       ::= { ippmAggrMeasureEntry 3 }
```

--

```
   -- ippmReport  Group
   --

   --
   --
   -- ippmReportSetupTable
   --
   --

   ippmReportSetupTable OBJECT-TYPE
          SYNTAX     SEQUENCE OF IppmReportSetupEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "The ippmReportSetupTable is a list of definition of reports.
It
                  defines the results of a network or aggregated measures that
are
                  to be reported. A report is saved in the ippmReportTable, or
sent
                  to an application using a SNMP Trap, a SNMP inform PDU, an
email
                  or a SMS. The reporting task is not intended to be a batch
action
                  processed at the end of the measure. It is coupled with
threshold
                  detections and event filtering to deliver application level
                  events and data, while preserving scalability.

                  It extends the definition of a measure: the definition of a
                  measure may include the definition of a report."
          ::= { ippmReport 1 }

   ippmReportSetupEntry OBJECT-TYPE
          SYNTAX     IppmReportSetupEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "The report applies to the results of the measure which is
                  extended by the current report definition.

                  Typically the creation of a report sets both the values of
the
                  new measure and those of the new IppmReportSetupEntry.
                  The ippmReportSetupDefinition describes the data and the
events
                  to include in the report. The definition consists in a list
of
```

```
                  tasks to perform on the results of the measure.

                  A report is associated to a network measure or to an
aggregated
                  measure.

                  Note 1 : To associate a report to an existing measure the
manager
                  suspends the measure while setting the ippmMeasureStatus to
                  'notInService'. Then he setups the report fields and
activates
                  the measure while setting the ippmMeasureStatus to 'active'.

                  Note 2 : A report is tied to a measure. The period of the
measure
                  "
```

```
        INDEX { ippmMeasureOwner, ippmMeasureIndex }
        ::= { ippmReportSetupTable 1 }

   IppmReportSetupEntry ::=
        SEQUENCE {
                ippmReportSetupDefinition             IppmReportDefinition,
                ippmReportSetupMetricThreshold        Integer32,
                ippmReportSetupDurationThreshold      Integer32,
                ippmReportSetupNMS                    SnmpAdminString,
                ippmReportSetupNotification           OBJECT IDENTIFIER,
                ippmReportSetupStatus                 RowStatus
        }


   ippmReportSetupDefinition OBJECT-TYPE
        SYNTAX IppmReportDefinition
        MAX-ACCESS read-create
        STATUS     current
        DESCRIPTION
                "The description of the events and actions that are used in
the
                definition of the report.
                Send the report using the type of message selected by the
bits 8
                to 12. The report consists of the results of the measure
which
                have been saved in the ippmReportTable. If the
                onEventSendReport(7) bit is unset, the report is not saved.

                The message sent is a notification defined in the
                ippmNotifications node. The notification sent depends on the
step
                of the measure:
                + Singleton events are sent using the notification
                ippmSingletonAlarm
                + Exceeded events durations are sent using the notification
                ippmEventsDurationExceededAlarm
                + A report of a cycle of measure is sent using the
notification
                ippmCycleOfMeasureReport
                + A report of a complete measure is sent using the
notification
                ippmCompletedMeasureReport

                Example 1:
                The report setup of an alarm to be sent to the owner in a
SNMP
                Trap each time the two results are found on each side of the
```

metric threshold value of 5:

```
ippmReportSetupMetricThreshold 5
ippmReportSetupDefinition {
onSingleton(1),
reportOnlyUptoDownMetricResults(4),
inSNMPTrapPDU(8)
}
```

Example 2:

```
                  The setup of a report to be sent to the owner in a SNMP
                  informRequestPDU per measure cycle. It reports the two
results
                  found on each side of the metric  threshold of 5:

                  ippmReportSetupMetricThreshold 5
                  ippmReportSetupDefinition {
                  onMeasureCycle(2),
                  reportOnlyUptoDownMetricResults(4),
                  inInformRequestPDU(10),
                  onReportDeliveryClearHistory(13)
                  }

                  Default report:
                  The default report provides the control protocol with an
implicit
                  mechanism to forward the result of a cycle of measure to the
                  owner of the measure while deleting the results corresponding
to
                  this cycle of measure from the ippmHistoryTable on reception
of
                  the response to the InformRequestPDU :
                  ippmReportSetupDefinition {
                  onMeasureCycle(2),
                  inInformRequestPDU(10),
                  onReportDeliveryClearHistory(13)
                  }
                  "
          DEFVAL { { onMeasureCycle, inInformRequestPDU,
          onReportDeliveryClearHistory} }
          ::= { ippmReportSetupEntry 1 }


   ippmReportSetupMetricThreshold OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "An event is generated when the result of the measure exceeds
the
                  value of ippmReportSetupMetricThreshold.
                  The threshold has the same unit as the metric. The metric
unit is
                  recorded in the object ippmMetricsUnit of this metric entry
in
                  the ippmMetricTable.
                  "
          ::= { ippmReportSetupEntry 2 }
   ippmReportSetupDurationThreshold OBJECT-TYPE
          SYNTAX Integer32
          UNITS      "Seconds"
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "An event is generated when contiguous results of the measure
are
                  over the ippmReportSetupMetricThreshold, during
                  ippmReportSetupDurationThreshold seconds.
                  Performance:
                  To improve the performance the
ippmReportSetupDurationThreshold
                  may have the same value as the ippmMeasurePeriod.
                  The default value of ippmReportSetupDurationThreshold is
                  ippmMeasurePeriod. That improves the performance because the
                  threshold comparison is synchronized with the
ippmMeasurePeriod
                  aggregation cycle. That improves the performance because it
                  synchronized the report exportation with the management of
the
                  history and report records of a measure."

          DEFVAL { 15 }
          ::= { ippmReportSetupEntry 3 }

   ippmReportSetupNMS OBJECT-TYPE
          SYNTAX SnmpAdminString
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  "The recipient of the report may be provided in the setup. By
                  default the recipient of the report is the owner of the
measure.
                  Its addresses are recorded in the ippmOwnersTable.
                  The type of ippmReportSetupNMS is not InetAddress because the
                  report may be sent using SMS or fax.
                  "
          ::= { ippmReportSetupEntry 4 }

   ippmReportSetupNotification OBJECT-TYPE
          SYNTAX     OBJECT IDENTIFIER
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
                  " ippmReportSetupNotification identifies the notification
used to
                  send the report.  The definition of the notification defines
the
                  content and the format of the report. "
          ::= { ippmReportSetupEntry 5 }

   ippmReportSetupStatus OBJECT-TYPE
          SYNTAX     RowStatus
          MAX-ACCESS read-create
          STATUS     current
          DESCRIPTION
```

```
                 "The status of this table entry. "
         ::= { ippmReportSetupEntry 6 }


--
-- ippmReportTable
--
```

```
   ippmReportTable OBJECT-TYPE
          SYNTAX     SEQUENCE OF IppmReportEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "The ippmReportTable logs the results of the reports. The
results
                  consist of a subset of the results of a measure as described
in
                  the report definition. The activation of an up and down
filtering
                  in the report definition limits the results logged to those
                  corresponding to major events. Otherwise, the ippmReportTable
is
                  identical to the ippmHistoryTable."

          ::= { ippmReport 2 }


   ippmReportEntry OBJECT-TYPE
          SYNTAX     IppmReportEntry
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "A report is a list of results of a measure. This sample is
                  associated with the ippmReportSetupEntry which has set up the
                  report. An ippmReportEntry entry is one of the results of a
                  measure to report.

                  An ippmReportEntry entry is one of the results of a measure
                  identified by ippmReportOwner, ippmReportIndex,
ippmReportIndex
                  and ippmHistoryIndex.

                  In the index:

                  + ippmMeasureOwner identifies the owner of the measure;

                  + ippmMeasureIndex identifies the measure in the owner
namespace;

                  + ippmMetricIndex identifies the metric measured in
                  ippmMetricTable;

                  + ippmReportIndex is the local index of the result on the
report
                  table."

          INDEX { ippmMeasureOwner, ippmMeasureIndex, ippmMetricIndex,
          ippmReportIndex }
          ::= { ippmReportTable 1 }

   IppmReportEntry ::=
          SEQUENCE {
                  ippmReportIndex             Integer32,
                  ippmReportSequence          Integer32,
                  ippmReportTimestamp         GMTTimeStamp,
                  ippmReportValue             Integer32
       }

   ippmReportIndex OBJECT-TYPE
          SYNTAX Integer32 (1.. 65535)
          MAX-ACCESS not-accessible
          STATUS     current
          DESCRIPTION
                  "The local index of the result of a metric measure"
          ::= { ippmReportEntry 1 }

   ippmReportSequence OBJECT-TYPE
          SYNTAX Integer32 (1.. 65535)
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION

                  " ippmReportSequence is the sequence index of the measurement
                  results of the measure of a metric.

                  Network metrics:

                  It's the sequence index of a measurement packet. Typically,
it
                  identifies the order of the packet in the stream of packets
sends
                  by the source.

                  Aggregated metrics:

                  It is the sequence index of the aggregated metric results
                  computed."
          ::= { ippmReportEntry 2 }



   ippmReportTimestamp OBJECT-TYPE
          SYNTAX GMTTimeStamp
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "The instant of the measure of the result."
          ::= { ippmReportEntry 3 }

   ippmReportValue OBJECT-TYPE
          SYNTAX Integer32
          MAX-ACCESS read-only
          STATUS     current
          DESCRIPTION
                  "The value."
```

```
        ::= { ippmReportEntry 4 }
```

```
   ippmOnReportFullAction   OBJECT-TYPE
          SYNTAX  INTEGER {
                  wrap(1),
                  suspend(2),
                  resume(3)
          }

          MAX-ACCESS read-write
          STATUS     current
          DESCRIPTION
                  "Action to take when the report log is full. The user 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 report log, and
assumes
                  the data has been cleared."

          ::= { ippmReport 3 }

   --
   -- IPPM  Notifications
   --


   ippmSingletonAlarm    NOTIFICATION-TYPE
          OBJECTS      {
                  ippmMetricUnit,
                  ippmReportTimestamp,
                  ippmReportValue
          }
          STATUS       current
          DESCRIPTION
                  "A notification sent because 2 contiguous results are on
opposite
                  sides of the metric threshold value.

                  The notification contains the instances of the
ippmReportValue
                  object that exceeded the threshold.
                  The notification contains the instances of the
                  ippmReportTimestamp identifying the time the event occurred."
          ::= { ippmNotifications 1 }

   ippmEventsDurationExceededAlarm    NOTIFICATION-TYPE
          OBJECTS      {
                  ippmMetricUnit,
                  ippmReportTimestamp,
                  ippmReportValue
          }
          STATUS       current
          DESCRIPTION
```

```
                "A notification sent when the duration of contiguous raising
                ippmReportSetupMetricThreshold exceeds the
                ippmReportSetupDurationThreshold value.

                The notification contains the instances of the
ippmReportValue
                object that exceeded the threshold.

                The notification contains the instances of the
                ippmReportTimestamp identifying the time the event occurred."
        ::= { ippmNotifications 2 }


   ippmCycleOfMeasureReport    NOTIFICATION-TYPE
        OBJECTS      {
                ippmMetricUnit,
                ippmHistoryTimestamp,
                ippmHistoryValue
        }
        STATUS       current
        DESCRIPTION
                "A notification sent when a measure cycle completes.

                The notification contains the instances of the
ippmReportValue
                objects saved in the ippmReportTable for this measure cycle.
The
                ippmHistoryTimestamp of the index identifies the time the
                measures where performed."
        ::= { ippmNotifications 3 }

   ippmCompletedMeasureReport    NOTIFICATION-TYPE
        OBJECTS      {
                ippmMetricUnit,
                ippmHistoryTimestamp,
                ippmHistoryValue
        }
        STATUS       current
        DESCRIPTION
                "A notification sent when a measure completes.
                The index of the included ippmReportSetupDefinition object
                identifies the ippmMeasureEntry and the ippmResultSetupEntry
that
                specified the report.

                The notification contains the instances of the
ippmReportValue
                objects saved in the ippmReportTable for this measure cycle.
```

```
The
                  ippmHistoryTimestamp of the index identifies the time the
                  measures where performed."
          ::= { ippmNotifications 4 }

   ippmHistoryLogFull    NOTIFICATION-TYPE
          OBJECTS      {
                  ippmOnHistoryFullAction
          }
          STATUS       current
          DESCRIPTION
                  "A notification sent when the history log is full. It
indicates
                  what action is to be taken. If the action is wrap the agent
will
                  write over existing records in the beginning of the log file.
If
                  the action is suspend, the agent halts all recording of
measures
                  in the history table. If the action is resume, the agent
begins
                  writing measures again in the history log"
          ::= { ippmNotifications 5 }


   ippmReportLogFull    NOTIFICATION-TYPE
          OBJECTS      {
                  ippmOnReportFullAction
          }
          STATUS       current
          DESCRIPTION
                  "A notification sent when the report log is full. It
indicates
                  what action is to be taken. If the action is wrap the agent
will
                  write over existing records in the beginning of the log file.
If
                  the action is suspend, the agent halts all recording of
measures
                  in the report table. If the action is resume, the agent
begins
                  writing measures again in the report log"
          ::= { ippmNotifications 6 }



   --
   -- 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, ippmMeasureGroup, ippmNetworkMeasureGroup,
                  ippmHistoryGroup,  ippmAggrMeasureGroup, ippmReportGroup,
                  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, ippmMeasureGroup, ippmNetworkMeasureGroup,
                  ippmHistoryGroup,  ippmAggrMeasureGroup, ippmReportGroup,
                  ippmNotificationGroup
          }
          GROUP ippmOwnersGroup
          DESCRIPTION
                  "The ippmOwnersGroup is needed if VACM is not implemented."
          ::= { ippmCompliances 2 }


   ippmProbeCompliances         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, ippmMeasureGroup, ippmNetworkMeasureGroup,
                  ippmHistoryGroup
          }
          ::= { ippmCompliances 3 }


   ippmSystemGroup    OBJECT-GROUP
          OBJECTS {
                  ippmSystemSynchronizationDesc,
                  ippmSystemTime,
                  ippmSystemSynchronizationType,
                  ippmSystemClockResolution,
                  ippmSystemCurrentSynchronization,
                  ippmSynchronizationTime,
                  ippmSynchronizationStratum,
                  ippmSynchronizationResolution,
                  ippmPointOfMeasureMgmtAddrType,
                  ippmPointOfMeasureMgmtAddress,
                  ippmPointOfMeasureTypePAddress,
                  ippmPointOfMeasureAddress
          }
          STATUS  current
          DESCRIPTION
                  "The IPPM  System Group"
```

```
       ::=  { ippmGroups  1}

ippmMeasureGroup  OBJECT-GROUP
       OBJECTS  {
               ippmMetricCapabilities,
               ippmMetricUnit,
               ippmMetricDescription,
               ippmMetricMaxHistorySize,
               ippmMeasureName,
               ippmMeasureMetrics,
               ippmMeasureBeginTime,
               ippmMeasureClockPeriodUnit,
               ippmMeasureClockPeriod,
               ippmMeasureDurationUnit,
               ippmMeasureDuration,
               ippmMeasureHistorySize,
               ippmMeasureStorageType,
               ippmMeasureStatus
       }
       STATUS  current
       DESCRIPTION
               "The IPPM Measure Group"
       ::=  { ippmGroups  2}

ippmNetworkMeasureGroup    OBJECT-GROUP
       OBJECTS  {
               ippmNetworkMeasureSrcTypeP,
               ippmNetworkMeasureSrc,
               ippmNetworkMeasureDstTypeP,
               ippmNetworkMeasureDst,
               ippmNetworkMeasureClockPattern,
               ippmNetworkMeasurePoissonRate,
               ippmNetworkMeasureTimeoutDelay,
               ippmNetworkMeasureL3PacketSize,
               ippmNetworkMeasureDataPattern
       }
       STATUS  current
       DESCRIPTION
               "The IPPM Network Measure Group"
       ::=  { ippmGroups  3}


ippmHistoryGroup   OBJECT-GROUP
       OBJECTS  {
               ippmHistorySequence,
               ippmHistoryTimestamp,
               ippmOnHistoryFullAction,
               ippmHistoryValue
       }
       STATUS  current
       DESCRIPTION
               "The IPPM History Group"
       ::=  { ippmGroups  4}
```

```
ippmAggrMeasureGroup       OBJECT-GROUP
       OBJECTS  {
```

```
                ippmAggrMeasureHistoryOwner,
                ippmAggrMeasureHistoryOwnerIndex,
                ippmAggrMeasureHistoryMetric
        }
        STATUS  current
        DESCRIPTION
                "The IPPM AggregatedMeasure Group"
        ::=  { ippmGroups  5}

ippmReportGroup    OBJECT-GROUP
        OBJECTS  {
                ippmReportSetupDefinition,
                ippmReportSetupMetricThreshold,
                ippmReportSetupDurationThreshold,
                ippmReportSetupNMS,
                ippmReportSetupNotification,
                ippmReportSetupStatus,
                ippmReportSequence,
                ippmReportTimestamp,
                ippmReportValue,
                ippmOnReportFullAction
        }
        STATUS  current
        DESCRIPTION
                "The IPPM Report Group"
        ::=  { ippmGroups  6}

ippmOwnersGroup    OBJECT-GROUP
        OBJECTS  {
                ippmOwnersOwner,
                ippmOwnersGrantedMetrics,
                ippmOwnersGrantedRules,
                ippmOwnersIpAddress,
                ippmOwnersEmail,
                ippmOwnersSMS,
                ippmOwnersStatus,
                ippmOwnersIpAddressType,
                ippmResultSharingMeasureOwner,
                ippmResultSharingMeasureIndex,
                ippmResultSharingGrantedOwner,
                ippmResultSharingStatus
        }
        STATUS  current
        DESCRIPTION
                "The IPPM Owners Group"
        ::=  { ippmGroups  7}
```

```
ippmNotificationGroup       NOTIFICATION-GROUP
       NOTIFICATIONS  {
               ippmSingletonAlarm,
               ippmCycleOfMeasureReport,
               ippmCompletedMeasureReport,
               ippmEventsDurationExceededAlarm,
               ippmHistoryLogFull,
               ippmReportLogFull
       }
       STATUS  current
       DESCRIPTION
               "The IPPM Notification Group"
       ::=  { ippmGroups  8}


END
```

9. Security Considerations

9.1. 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.

9.2. 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.

9.3. 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.

10. Document management

10.1. Open issues

Describe incompatible bit combinations in IPPMreport and granted metric

Run SMIlint.

Discussion on the management of the history size.

10.2. changes since release 00

+ Put in a description of the relationship of certain tables, particularly the measure/network measure/aggregated measure table.

+ The TC GMTTimeStamp is the common type to define timestamp objects.

+ ippmHisoryTable index simplified: ippmHistoryTimestamp replaced with ippmHistorySqceNdx in the index.

+ The MIB has been compiled using net-snmp.

+ Snmpadminstring replaces Displaystring.

+ IP addresses defined using INETaddresstype.

+ Sharing table is optional to permit the VACM framework to be used.

+ The description of the network measure table emphases that the set up of network measure is not permitted using SNMP.

+ The TC StandardMetrics is removed and replaced with the table ippmMetricsTable.

+ The table pointOfMeasureTable is added to describe multiples interfaces devices

+ 5 tables have been changed to read/create: ippmOwnersTable, ippmMeasureTable, ippmAggrMeasureTable, ippmResultSharingTable, and ippmReportSetupTable.

+ IppmHistoryTable and ippmReportTable index reviews:

IppmHistorySqceNdx field added in the ippmHistoryTable.
INDEX modified. IppmHistorySqceNdx replaces IppmHistoryTimemark.

+ IppmSystem group refurbished:
  IppmSystemTimer renamed ippmSystemTime.
  Current and last synch event concept generalized in the ippmSynchronizationTable.

10.3. Changes since release 01

+ Document Format:
  Make use of the regular MIB object indentation.

+ Typos correction: ippmMeasureHystorySize and so on.

+ Time unit textual convention:
  Enumerations listed in description clauses (e.g. ms, us, ns may not be universally understood so explicitly named as millisecond, microsecond, nanosecond)

+ Clarify ClearHistory and ClearReport definition:
  OnReportDeliveryClearHistory and OnReportDeliveryClearReport options

+ Added scalars ippmOnReportFullAction and ippmOnHistoryFullAction:
  To take action when the tables are full. A scalar, which is read-write and indicates the action to be taken when the log is full.
  Options are: wrap, suspend, resume. Same was done for report group.

+ Conformance section:
  Added the MODULE-COMPLIANCE macro and the corresponding OBJECT-GROUPS instances.
  Added a compliance instances for proxy mode, proxy inter-domain mode and probe mode.

+ PointOfMeasure:
  Put in ippmPointOfMeasureMgmtAddrType-> InetAddressType with ippmPointOfMeasureMgmtAddress-> InetAddress.
  Changed point of measure address to be INET also.

```
      + Took out default point of measure address:
              Added OwnersIpAddressType to be in pair with OwnersIpAddress

      + Added ippmSynchronizationResolution in the ppmSynchronizationTable:
              It indicates the new time resolution (Henk request).

      + Added an object ippmReportSetupNotification in the report setup.

      + IppmHistoryIndex added in the history table:
```

To differentiate the result index from the test packet order.

+ IppmReportIndex added in the report table:
To differentiate the result index from the test packet order.

+ Smilint: with the option -s -l6:
Name length exceeded 32 chars:
Prefix:
+ ippmAggregatedMeasure -> ippmAggrMeasure;
+ IppmSystemSynchronizationDescription
-> ippmSystemSynchronizationDescr;
+ IppmReportSetupEventsDurationThreshold
-> ippmReportSetupDurationThreshold.

+ ippmNotifications identified under ippm

+ TC OwnerString replaced with IppmOwnerString to fix a warning
of the key length;

+ Gain 0 error and warning !

+ ippmAggrMeasureStatus removed:
The status of the row is managed in the ippmMeasureTable

+ Notifications:
definition clarified;
ippmReportTimestamp added to notification
ippmEventsDurationExceededAlarm, ippmSingletonAlarm,
ippmCycleOfMeasureReport, ippmCompletedMeasureReport.

+ IppmNetworkMeasureEntry :
ippmNetworkMeasurePoissonRate added as the average rates.

+ TypeP redefined as a SnmpAdminString instead of a raw OCTET STRING
e.g: '0800000800000000011020000'H -> "ip.ipip4".
open issue:
is there a need to indicate the number of parameters of the
protocol identifier ? "ip.ipip4.2" or "ip.ipip4" ?

+ TypePaddress Textual convention created:
Dst and Src value is a display string instead of a raw OCTET
STRING. It is the list of parameters of a TypeP.
e.g:
Src address TypeP is "ip.ipip4": 128.2.6.7 in 192.168.1.1.
Src value was '0A04C0A801010480020607'H.

Src is now "192.168.1.1 128.2.6.7".

open issue:

is there any potential parameter with one or more space
inside ?

11. References

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

[2] Mahdavi J. and V. Paxson, "IPPM Metrics for Measuring Connectivity", RFC 2678, September 1999.

[3] Almes, G., Kalidindi, S. and M. Zekauskas, "A One-way Delay Metric for IPPM", RFC 2679, September 1999.

[4] Almes, G., Kalidindi, S. and M. Zekauskas, "A One-way Packet Loss Metric for IPPM", RFC 2680, September 1999.

[5]Almes, G., Kalidindi, S. and M. Zekauskas, "A Round-trip Delay Metric for IPPM.", RFC 2681, September 1999.

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

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

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

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

[10] 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.

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

[12] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M., and S. Waldbusser, "Conformance Statements for SMIv2", STD 58,

RFC 2580, April 1999.

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

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

[15] 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.

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

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

[18] 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.

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

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

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

12. Acknowledgments

A Kerbe.

13. 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