Network Working Group E. Stephan/J. Jewitt

Internet Draft France Telecom R&D

Document: draft-ietf-ippm-reporting-mib-00.txt June 20, 2002

IPPM reporting MIB

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [1].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

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 the measures using the IP performance metrics specified by the IPPM Working Group.

It specifies the objects to manage the results of the measure of metrics standardized by IPPM Working Group. They are built on notions introduced and discussed in the IPPM Framework document, RFC 2330 [2].

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 in 3 major components:

A general framework for defining performance metrics, 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 [3]; The One-way Delay Metric for IPPM, RFC 2679 [4]; The One-way Packet Loss Metric for IPPM, RFC 2680 [5]; The Round-trip Delay Metric for IPPM, RFC 2681 [6].

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 [7].

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 [8], STD 16, RFC 1212 [9] and RFC 1215 [10]. The second version, called SMIv2, is described in STD 58, RFC 2578 [11], STD 58, RFC 2579 [12] and STD 58, RFC 2580 [13].

Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [14]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [15] and RFC 1906 [16]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [16], RFC 2572 [17] and RFC 2574 [18].

Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [14]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [19].

A set of fundamental applications described in RFC 2573 [20] and the view-based access control mechanism described in RFC 2575 [21].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [22].

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 results of these metrics. This memo defines a Management Information Base for managing the results of the measures 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 [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 [23],[24] 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 measures and requires common references for time and for measure identification. The IPPM-REPORTING-MIB defines an absolute timeFilter.

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 measure 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 a measure. The memo lists them, while also looking for a way to integrate them with the IPPM-REPORTING-MIB . This section is informational, but helps 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 protocol and test protocol. 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 access by SNMP applications. They are not in constant contact with the measurement devices. For this reason, this MIB allows continuous measures collection and statistics computation.

The objects defined in this document are not intended for direct manipulation..

4.1. Textual Conventions

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

4.1.1. Packet of type P

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 configure finely and unambiguously the type P of the packets, and the parameters of the protocol suites of the type P.

RMON2 introduced the concept of protocol identifiers. The RFC2895 [25] specifies a macro for the definition of protocol identifier. The RFC2896 [26] 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 as a 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 protocol identifier encoding rule, but without trailing parameters.

The packet encapsulation defined in an instance of TypeP embeds the format of "Src" and "Dst" and their values. These addresses type and value depend on the type P of the packet, IP version 4, V6, IP in IP... Both participate to 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 8.0.0.8.0.0.0.0.17.2.0.0 (ip.ipip4). The trailer 2.0.0 in the TypeP indicates that there are 2 parameters. In the IPPM context these 2 parameters are provided in Src, which has the value 10.4.192.168.1.1.4.128.2.6.7.

4.1.2. GMTDateAndTime

This textual convention defines the date and time, and is represented by the following format: year, month, day, hour, minutes, seconds, fractions of second.

The first part is human readable. The fields year, month, day, hour, minutes are seconds are printable characters.

The last field is the fraction of second. The fraction step is 250 picoseconds.

or example, 50 ms is 200000000 times 250 picosecond which correspond to 0BEBC200'H. 0001000201090200010501000BEBC200 represent 8:15pm, 10 econds and 50 ms GMT on 19 February 2001 and is displayed as 01-02-9,20:15:10, 200000000.

4.1.3. GmtTimeFilter

GmtTimeFilter uses an absolute reference of time, and is intended to be used for the index of a table. It allows an application to download only those rows changed since a particular time. A row is considered changed if the value of any object in the row changes, or if the row is created or deleted.

Each new conceptual row will be associated with the timeMark instance that was created at the value of ippmTimeSysTimer.

It is intended to provide an absolute timestamp index for the results of measures. Typically for each singleton produced by an IPPM measure is associated the timemark corresponding to the moment of the measure.

Illustrations:

Consider the 2 tables measureTable and resultTable measureTable OBJECT-TYPE

```
SYNTAX     SEQUENCE OF MeasureEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION ''
::= { fooMib 1 }
INDEX { measureIndex }

resultTable OBJECT-TYPE
SYNTAX     SEQUENCE OF ResultEntry
MAX-ACCESS not-accessible
STATUS     current
DESCRIPTION ''
::= { fooMib 2 }
INDEX { measureIndex, resultTimeMark }

ResultEntry {
   resultTimeMark  TimeFilter,
   resultCounts    Counter
}
```

LetÆs assume there are two measure rows in the measure table (measureIndex == 1, measureIndex == 2) which produced results asynchronously (e.g. made at Poisson intervals or sibling) and logged them in the result table.

LetÆs also assume that Measure 1 produced the result 34 at time 00010002010902000105010000BEBC200 GMT, while Measure 2 produced the value 54 **10ms later at time 0001000201090200010501000E4E1C00 GMT, and that** both rows are updated on several later occasions such that the current values are 37 and 53 respectively.

Then the following resultCounts instances would exist:

```
   resultCounts.1.0001000201090200010501000BEBC200 34
   resultCounts.2.0001000201090200010501000E4E1C00 54
   resultCounts.1.00010002010902000105010016A65700 65
   resultCounts.1.0001000201090200010501000E4E1C00 57
   resultCounts.2.00010002010902000105010001312D000 48
   resultCounts.2.0001000201090200010501001443FD00 53
   resultCounts.1.00010002010902000105010101312D00 49
   resultCounts.1.000100020109020001050101 04C4B400 37
```

The following get-bulk explains how an NMS retrieves the results of the measures.

```
get-bulk(nonRptrs=1, maxReps=10, resultCounts.1);
returns:
        resultCounts.1. 0001000201090200010501000BEBC200 == 34
```

```
resultCounts.1.0001000201090200010501001 6A65700 == 65
```

```
        resultCounts.1.0001000201090200010501010l312D00 == 49
        resultCounts.1.00010002010902000105010104C4B400 == 37
        # return lexigraphically-next two MIB instances

get-bulk(nonRptrs=0, maxReps=2,
resultCounts.1.00010002010902000105010000E4E1C00 ,
resultCounts.2.00010002010902000105010000E4E1C00 );
returns:
        resultCounts.1.00010002010902000105010016A65700 == 65
        resultCounts.2.00010002010902000105010001312D000 == 48
        resultCounts.1.00010002010902000105010000E4E1C00 == 57
        resultCounts.2.000100020109020001050100001443FD00 == 53


get-bulk(nonRptrs=0, maxReps=2,
resultCounts.1.00010002010902000105010104C4B400 ,
resultCounts.2.00010002010902000105010104C4B400 );
returns:
        return lexigraphically-next two MIB instances
        no 'resultTable' counter values at all are returned because
neither counter has been updated since the time
00010002010902000105010104C4B400
```

4.1.4. Report definition

A report consists of sending or logging a subset of results of measure. The elaboration of the report consists of actions to perform 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 measure 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 an NMS address.
+ 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.

4.1.5. IppmStandardMetrics

The TEXTUAL-CONVENTION IppmStandardMetrics defines the standardized IPPM metrics.

4.2. Structure of the MIB

The MIB is arranged as follow:

- ippmOwnersGroup
- ippmSystemGroup
- ippmMeasureGroup
- ippmHistoryGroup
- ippmNetworkMeasureGroup
- ippmAggregatedMeasureGroup
- ippmReportGroup
- ippmNotifications

4.2.1. The ippmOwners Group

This group controls access to the probe.

4.2.2. The ippmSystem Group

This group consists of a set of parameters describing the clock synchronization over the 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, ippmMeasureTable.

The measurement entity describes in the ippmMetricsTable of the SNMP agent the local implementation of the standardized metrics.

The control protocol registers a description of the existing measures in the ippmMeasureTable and in the auxiliary measure tables.

ippmMeasureTable holds the common part of a measure, while the specific parameters are handled in the corresponding auxiliary table (ippmNetworkMeasure, ippmAggregatedMeasureTableà) .

The results of the measures 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 ippmAggregatedMeasure Group

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

As the OBJECT IDENTIFIER, which identifies the instance, begins with the owner value, the remaining value 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 exist 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, s 2 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 NMS applications. An instance of an object managed by an NMS is identified by the NMS OwnerString and the private index provided by the NMS.

As the NMS 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.

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.

Also, in this mode, it is recommended to implement the IPPM-REPORTING-MIB Measure table in read only in order to allow an NMS to read the status of their measures.

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 with the control and test protocols

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 measurement 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 Ipv4 best effort.

8. Definition

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

   IMPORTS
        MODULE-IDENTITY,
        NOTIFICATION-TYPE,
        OBJECT-TYPE,
        Integer32,
        Counter32,
        experimental
                FROM SNMPv2-SMI
        OwnerString
                FROM RMON-MIB
        protocolDir
                FROM RMON2-MIB
        DisplayString,
        TimeStamp,
        DateAndTime,
        TruthValue,
        RowStatus,
        StorageType,
        TEXTUAL-CONVENTION
                FROM SNMPv2-TC
        MODULE-COMPLIANCE,
        OBJECT-GROUP,
        NOTIFICATION-GROUP
                FROM SNMPv2-CONF;


   ippmReportingMib MODULE-IDENTITY
       LAST-UPDATED "200202011200Z"    -- March 17, 2002
```

```
        ORGANIZATION    "France Telecom - R&D"
        CONTACT-INFO
```

```
        "Mail : Emile Stephan
        France Telecom - R&D, Dpt. CPN
        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.
   "
        ::= { ippm 2 }
   --
   -- TEXTUAL-CONVENTION
   --


   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),
                ms(7),
                us(8),
                ns(9)
        }
   --
   --
   -- A absolute, GMT timer using UTC like convention.
   --
```

```
--

GMTDateAndTime ::= TEXTUAL-CONVENTION
```

```
      DISPLAY-HINT "d-d-d,d:d:d,4d"
      STATUS       current
      DESCRIPTION
               "A date-time specification.

               field  octets  contents                  range
               -----  ------  --------                  -----
               1        1-2    year*                   0..99
               2        3-4    month                   1..12
               3        5-6    day                     1..31
               4        7-8    hour                    0..23
               5        9-10   minutes                 0..59
               6        11-12  seconds                 0..59
               7        13-16  250 picoseconds         0..2^32-1
               "
      SYNTAX       OCTET STRING (SIZE (16))


   GmtTimeFilter ::= TEXTUAL-CONVENTION
       STATUS        current
       DESCRIPTION
               "GmtTimeFilter TC is inspired by the TimeFilter defined
               in  RMON2. The reference for the time of TimeFilter is
               the local value of sysUptime, while GmtTimeFilter uses
               an absolute reference of
time.Æ                                              Æ

      SYNTAX    GMTDateAndTime


   TypeP  ::= TEXTUAL-CONVENTION
      DISPLAY-HINT "1d."
      STATUS       current
      DESCRIPTION
               "This textual convention is 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 for the definition of protocol identifiers while
               its companion document, the RFC2896 defines a set of
               protocol identifiers.

               Notes: An IPPM TypeP does not differ from RMON2 Protocol
               identifiers, but TypeP usage in IPPM MIB differs from
```

Protocol identifier usage in RMON2. A IPPM measure is
active, so generally TypeP 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.

            For example, the RFC2896 defines the protocol identifier
            '16.0.0.0.1.0.0.8.0.0.0.0.6.0.0.0.23.3.0.0.0' which
            represents ether2.ip.tcp.telnet and the protocol
            identifier 16.0.0.0.1.0.0.8.0.0.0.0.4.0.0.0.17.3.0.0.0
            which stands for ether2.ip.ipip4.udp. The corresponding
            TypeP are '12.0.0.8.0.0.0.0.6.0.0.0.23.3.0.0.0'
            (ip.tcp.telnet) and 12.0.0.8.0.0.0.0.4.0.0.0.17.3.0.0.0
            (ip.ipip4.udp)."
     SYNTAX       OCTET STRING


   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 if 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.

        clearHistory(12):
               Remove all the results corresponding to this measure
                from the ippmHistoryTable.

        clearReport(13):
               Remove all the results corresponding to this measure
               from the ippmReportTable.
        "

        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),
                clearHistory(12),
                clearReport(13)
        }
```

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

```
      DESCRIPTION
      "The definition of the standardized IPPM metrics.
      If the draftMetrics bit is set then the other bits describe a WG
   draft metric identifier.
      "
      SYNTAX BITS {
              draftMetrics(0),
              instantaneousUnidirectionalConnectivity(1),
              instantaneousBidirectionalConnectivity(2),
              intervalUnidirectionalConnectivity(3),
              intervalBidirectionalConnectivity(4),
              intervalTemporalConnectivity(5),
              onewayDelay(6),
              onewayDelayPoissonStream(7),
              onewayDelayPercentile(8),
              onewayDelayMedian(9),
              onewayDelayMinimum(10),
              onewayDelayInversePercentile(11),
              onewayPacketLoss(12),
              onewayPacketLossPoissonStream(13),
              onewayPacketLossAverage(14),
              roundtripDelay(15),
              roundtripDelayPoissonStream(16),
              roundtripDelayPercentile(17),
              roundtripDelayMedian(18),
              roundtripDelayMinimum(19),
              roundtripDelayInversePercentile(20)
      }
   -- IPPM Mib objects definitions

   ippmCompliances      OBJECT IDENTIFIER       ::= { ippmMib 2 }
   ippmOwnersGroup      OBJECT IDENTIFIER       ::= { ippmMib 3 }
   ippmSystemGroup      OBJECT IDENTIFIER       ::= { ippmMib 4 }
   ippmMeasureGroup     OBJECT IDENTIFIER       ::= { ippmMib 5 }
   ippmHistoryGroup     OBJECT IDENTIFIER       ::= { ippmMib 6 }
   ippmNetworkMeasureGroup      OBJECT IDENTIFIER ::= { ippmMib 7 }
   ippmAggregatedMeasureGroup   OBJECT IDENTIFIER ::= { ippmMib 8 }
   ippmReportGroup              OBJECT IDENTIFIER ::= { ippmMib 9 }
   ippmNotifications            OBJECT IDENTIFIER ::= { ippmMib 10 }


   --
   -- ippmOwnersGroup
   --
   -- The ippmOwnersGroup 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 NMS entity wishing to create and activate remote Ippm
                measurements in an agent must previously be registered
                in the ippmOwnersTable.


                ippmOwnersTable content is read only.

                ippmOwnersTable is mandatory. It contains at least the
                owner 'monitor'.
                "

        ::= { ippmOwnersGroup 1 }

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

                For example, an instance of ippmOwnersOwner with an
                OwnerString '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. It is not used in anyway in the
                other IPPM tables."

        INDEX { ippmOwnersIndex }
        ::= { ippmOwnersTable 1 }

   IppmOwnersEntry ::= SEQUENCE {
        ippmOwnersOwner                  OwnerString,
        ippmOwnersIndex                  Integer32,
        ippmOwnersGrantedMetrics         IppmStandardMetrics,
        ippmOwnersGrantedRules           BITS,
        ippmOwnersIpAddress              DisplayString,
        ippmOwnersEmail                  DisplayString,
        ippmOwnersSMS                    DisplayString,
        ippmOwnersStatus                 OwnerString
   }

   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     OwnerString
      MAX-ACCESS read-only
      STATUS     current
      DESCRIPTION
              "The owner described by this entry."
      ::= { ippmOwnersEntry 2 }


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

 ippmOwnersGrantedRules OBJECT-TYPE
      SYNTAX     BITS {
              all(0),
              readonly(1),
              permanent(2),
              sender(2),
              receive(3),
              report(4),
              alarm(5)
 }
      MAX-ACCESS read-only
      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(2):
             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(4):
             The owner may setup aggregated metrics on the measures
corresponding to these metrics.

     alarm(5):
             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 }

ippmOwnersIpAddress OBJECT-TYPE
     SYNTAX     DisplayString
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
             "The IP address of the NMS corresponding to this owner.
The address is human readable and is represented using the dot
format."
     ::= { ippmOwnersEntry 5 }

ippmOwnersEmail OBJECT-TYPE
     SYNTAX     DisplayString
```

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

```
                "The email address of the NMS corresponding to this
   owner."
        ::= { ippmOwnersEntry 6 }

   ippmOwnersSMS OBJECT-TYPE
        SYNTAX     DisplayString
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The SMS phone number of the NMS corresponding to this
   owner."
        ::= { ippmOwnersEntry 7 }

   ippmOwnersStatus OBJECT-TYPE
        SYNTAX     RowStatus
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The status of this table entry."
        ::= { ippmOwnersEntry 8 }

--
--      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.

                IppmResultSharingTable is optional.
                IppmResultSharingTable content is read only.


                If this table is not implemented then the owner has only
                access to its measure results."

   ::= { ippmOwnersGroup 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 NMS.

             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          OwnerString,
     ippmResultSharingIndex          Integer32,
     ippmResultSharingMeasureOwner   OwnerString,
     ippmResultSharingMeasureIndex   Integer32,
     ippmResultSharingGrantedOwner   OwnerString,
     ippmResultSharingStatus         RowStatus
}
ippmResultSharingOwner OBJECT-TYPE
     SYNTAX OwnerString
     MAX-ACCESS read-only
     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 read-only
     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 OwnerString
        MAX-ACCESS read-only
```

```
     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-only
     STATUS     current
     DESCRIPTION
             "The index of the measure to be shared."
     ::= { ippmResultSharingEntry 4 }

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


--
-- ippmSystemGroup
--
--


ippmSystemTimer OBJECT-TYPE
     SYNTAX GMTDateAndTime
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
             "The current time of the system."
     ::= { ippmSystemGroup 1 }
```

```
   ippmSystemSynchonizationType OBJECT-TYPE
        SYNTAX INTEGER  {
                other(0),
                ntp(1),
                gps(2),
                cdma(4)
        }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "ippmSystemSynchonizationType 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(4)
                       The system is synchronized using the CDMA
                clocks.
                "
        ::= { ippmSystemGroup 2 }

   ippmSystemSynchonizationDescription OBJECT-TYPE
        SYNTAX DisplayString
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The description of the synchronization process."
        ::= { ippmSystemGroup 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 1/10 of
                millisecond. 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."

       ::= { ippmSystemGroup 4 }
```

```
   ippmSystemSynchronisationTime OBJECT-TYPE
        SYNTAX DateAndTime
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The time when the last synchronization of the clock
                occured. The last synchronization must be set even if
                the synchronization is not automatic."
        ::= { ippmSystemGroup 5 }

   ippmSystemClockAccuracy OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The most recent accuracy of the clock computed. The
                accuracy must be computed even if the synchronization is
                not automatic."
        ::= { ippmSystemGroup 6 }

   ippmSystemClockSkew OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The most recent skew of the clock computed. The
                ippmSystemSkew must be computed even if the
                synchronization is not automatic."
        ::= { ippmSystemGroup 7 }


   ippmSystemPrevClockAccuracy OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The previous accuracy of the clock measured. The
                ippmSystemPrevClockAccuracy must be computed even if the
                synchronization is not automatic."
        ::= { ippmSystemGroup 8 }

   ippmSystemPrevClockSkew OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The previous skew of the clock measured. The
```

```
                ippmSystemPrevClockSkew must be computed even if the
                synchronisation is not automatic."
        ::= { ippmSystemGroup 9 }
```

```
        ippmSystemSynchonizationOperStatus OBJECT-TYPE
        SYNTAX INTEGER  {
                other(0),
                unsynchronized(1),
                initializing(2),
                synchronized(4)
        }
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                " ippmSystemSynchonizationOperStatus describes the
                operational status of the clock synchronization.


                Other(0)
                The status of the synchronization is unknown.

                unsynchronized(1)
                The system is not synchronized.

                initializing(2)
                       The system is receiving synchronization
                information but is not yet synchronized.

                synchronized(4)
                       The system is synchronized.
                "
        ::= { ippmSystemGroup 10 }


   --

   --
   --
   -- ippmMeasureGroup
   --
   --
   --

   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 measure software handles
```

the table internally.

```
                ippmMetricTable is mandatory.
                ippmMetricTable content is read only.


                "
        ::= { ippmMeasureGroup 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     DisplayString,
                ippmMetricMaxHistorySize  Integer32
        }

   ippmMetricIndex OBJECT-TYPE
        SYNTAX Integer32 (1.. 65535)
        MAX-ACCESS read-only
        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
                "notImplemented
                        the metric is not implemented.
                implemented
                        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(6),
                kbyte(7),
                megabyte(8)
        }
        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 DisplayString
        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."
   ::= { 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
                ippmMeasureHystorySize.

                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 ippmMeasureHystorySize must not
                be over the value of ippmMetricMaxHistorySize
                corresponding to this metric in the ippmMetricTable.

                The ippmMeasureTable is mandatory.

                ippmMeasureTable content is read only. It means that the
                table is handled internally by the measurement
                software.
                "
        ::= { ippmMeasureGroup 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                OwnerString,
                ippmMeasureIndex                Integer32,
```

```
            ippmMeasureName                  DisplayString,
            ippmMeasureMetrics               IppmStandardMetrics,
```

```
            ippmMeasureBeginTime            GMTDateAndTime,
            ippmMeasureClockPeriodUnit      TimeUnit,
            ippmMeasureClockPeriod          Integer32,
            ippmMeasureDurationUnit         TimeUnit,
            ippmMeasureDuration             Integer32,
            ippmMeasureHystorySize          Integer32,
            ippmMeasureStorageType          StorageType,
            ippmMeasureStatus               RowStatus
        }

   ippmMeasureOwner OBJECT-TYPE
        SYNTAX     OwnerString
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The owner who has configured this entry."
        DEFVAL { "acme" }
        ::= { ippmMeasureEntry 1 }

   ippmMeasureIndex OBJECT-TYPE
        SYNTAX Integer32 (1.. 65535)
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The owner index of the measure. The value is managed by
                the owner."
        ::= { ippmMeasureEntry 2 }

   ippmMeasureName OBJECT-TYPE
        SYNTAX DisplayString
        MAX-ACCESS read-only
        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-only
        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."
          DEFVAL { { one-way-Delay, one-way-Packet-Loss } }
       ::= { ippmMeasureEntry 4 }


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


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

   ippmMeasureClockPeriod OBJECT-TYPE
       SYNTAX     Integer32
       MAX-ACCESS read-only
       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 ippmHistoryTimeMark result value of the
                metric computed corresponds to the ippmHistoryTimeMark
                value of the last metric result of the sequence used in
                the computation."
        DEFVAL { 60 }
        ::= { ippmMeasureEntry 7 }


   ippmMeasureDurationUnit OBJECT-TYPE
        SYNTAX TimeUnit

        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "Specifies the unit of the measure duration."
        DEFVAL { second }
        ::= { ippmMeasureEntry 8 }

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

   ippmMeasureHystorySize OBJECT-TYPE
        SYNTAX     Integer32
        MAX-ACCESS read-only
        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-only
          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-only
        STATUS     current
        DESCRIPTION
                "The status of this table entry. Once the entry status
                is set to active, the associate entry cannot be
                modified."
        DEFVAL { createAndGo }
        ::= { ippmMeasureEntry 12 }

   --
   -- ippmHistoryGroup
   --
   --

   --
   -- ippmHistoryTable
   --


   ippmHistoryTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF IppmHistoryEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
                "The table of the results of the measures."

        ::= { ippmHistoryGroup 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 the index members ippmMeasureOwner
                and ippmMeasureIndex.

                In the index :

                        + ippmMeasureOwner and ippmMeasureIndex identify
                the ippmMeasureEntry on whose behalf this entry was
                created
                        + ippmMetricIndex identifies the ippmMetricEntry
                of the metric to measure
                        + ippmLogTimeMark value is the absolute time
                when the result of the metric was measured.

                The ippmHistoryTimeMark value is the absolute time when
                the ippmHistoryValue was performed.

                IppmHistoryValue is the value of the metric measured at
                the time ippmHistoryTimeMark.

                Example:
                A one way delay measure is created by the entity 'acme'
                using the owner index 1 and setting the 6th bit
                (corresponding to One-way-Delay) of ippmMeasureMetrics
                to 1.
                Consider that the result of the one way delay measured
                for acme on the day 15 of June at 8h20mn 10s 50ms is 23.
                The result is identified as the singleton
                ippmHistoryValue.acme.1.6.00010002010902000010501000BEBC2
                00 and stored with value 23.

                Its value may be retrieved using a get-
                next(ippmHistoryValue.acme.1.6.00010002010902000010501000
                0000000) which returns
                (ippmHistoryValue.acme.1.6.00010002010902000010501000BEBC
                200 == 23). The ippmMetricIndex value of '6' corresponds
                to the 6th metric of ippmMetricTable which is Type-P-
                One-way-Delay.
                "
        INDEX { ippmMeasureOwner, ippmMeasureIndex, ippmMetricIndex,
   ippmHistoryTimeMark }
        ::= { ippmHistoryTable 1 }

   IppmHistoryEntry ::=
        SEQUENCE {
```

```
                ippmHistoryTimeMark     GMTDateAndTime,
                ippmHistorySqceNdx      Integer32,
```

```
                ippmHistoryValue        Integer32
        }
   ippmHistoryTimeMark OBJECT-TYPE
        SYNTAX GMTDateAndTime
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
        "The instant of the measure of the result."
        ::= { ippmHistoryEntry 1 }

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

                " ippmHistorySqceNdx 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 }

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

                "The value of the measure."
        ::= { ippmHistoryEntry 3 }


   --
   -- ippmNetworkMeasureGroup
   --


   --
   --
```

```
   -- 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. A network
                measure is a specific measure.

                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."
        ::= { ippmNetworkMeasureGroup 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 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."
        INDEX { ippmMeasureOwner, ippmMeasureIndex }
        ::= { ippmNetworkMeasureTable 1 }

   IppmNetworkMeasureEntry ::=
        SEQUENCE {
                ippmNetworkMeasureSrcTypeP              TypeP,
                ippmNetworkMeasureSrc                   OCTET STRING,
                ippmNetworkMeasureDstTypeP              TypeP,
                ippmNetworkMeasureDst                   OCTET STRING,
                ippmNetworkMeasureClockPattern          OCTET STRING,
```

```
             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 OCTET STRING
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "Specifies the address of the source of the measure.

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

                For example, if the ippmNetworkMeasureSrcTypeP indicates
                an encapsulation of 'ip', this object length is 4,
                followed by the 4 octets of the IP address, in network
                byte order."
        DEFVAL { '04C0210415'H } -- -> ip: 192.33.4.21
        ::= { 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."
        DEFVAL { '04000080001000'H } -- ->ip: 4.0.0.8.0.1.0
        ::= { ippmNetworkMeasureEntry 3 }

   ippmNetworkMeasureDst OBJECT-TYPE
        SYNTAX OCTET STRING
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "Specifies the address of the destination of the
                measure.
```

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

                For example, if the ippmNetworkMeasureDstTypeP indicates
                an encapsulation of 'ip', this object length is 4,
                followed by the 4 octets of the IP address, in network
                byte order."
        DEFVAL { '04C0220414'H } -- -> ip: 192.34.4.20
        ::= { 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.
                "
        DEFVAL { 11111111} -- 100% periodic
        ::= { ippmNetworkMeasureEntry 5 }

   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 6 }

   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 7 }

   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 8 }



   --
   --
   -- ippmAggregatedMeasureGroup
   --
   --
   --
   --
   -- ippmAggregatedMeasureTable
   --
   --

   ippmAggregatedMeasureTable OBJECT-TYPE
        SYNTAX     SEQUENCE OF IppmAggregatedMeasureEntry
        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."
        ::= { ippmAggregatedMeasureGroup 1 }

   ippmAggregatedMeasureEntry OBJECT-TYPE
        SYNTAX     IppmAggregatedMeasureEntry
        MAX-ACCESS not-accessible
        STATUS     current
        DESCRIPTION
        "Typically the configuration operation sets both the values of
```

the new ippmMeasureEntry and of the new
IppmAggregatedMeasureEntry.

```
        ippmAggregatedMeasureTable is mandatory.

        ippmAggregatedMeasureTable 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:
                + ippmAggregatedMeasureHistoryOwner,
                + ippmAggregatedMeasureHistoryOwnerIndex and
                + ippmAggregatedMeasureHistoryMetric

        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 }
        ::= { ippmAggregatedMeasureTable 1 }


   IppmAggregatedMeasureEntry ::=
        SEQUENCE {
                ippmAggregatedMeasureHistoryOwner       OwnerString,
                ippmAggregatedMeasureHistoryOwnerIndex  Integer32,
                ippmAggregatedMeasureHistoryMetric      Integer32
        }


   ippmAggregatedMeasureHistoryOwner OBJECT-TYPE
        SYNTAX OwnerString
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The owner of the measure to summarize. "
        ::= { ippmAggregatedMeasureEntry 1 }

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

   ippmAggregatedMeasureHistoryMetric OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
```

```
      STATUS     current
      DESCRIPTION
```

```
                "The metric of the measure to summarize. "
          ::= { ippmAggregatedMeasureEntry 3 }


   --
   -- ippmReportGroup
   --

   --
   --
   -- 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."
        ::= { ippmReportGroup 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."
        INDEX { ippmMeasureOwner, ippmMeasureIndex }
        ::= { ippmReportSetupTable 1 }

   IppmReportSetupEntry ::=
```

```
        SEQUENCE {
                ippmReportSetupDefinition       IppmReportDefinition,
```

```
                ippmReportSetupMetricThreshold  Integer32,
                ippmReportSetupEventsDurationThreshold  Integer32,

                ippmReportSetupNMS              DisplayString
        }

   ippmReportSetupDefinition OBJECT-TYPE
        SYNTAX IppmReportDefinition
        MAX-ACCESS read-only
        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 setup of an alarm to be sent to the owner in a SNMP Trap
   each time the staircase crosses 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 staircase values
   corresponding to a metric threshold of 5:

                ippmReportSetupMetricThreshold 5
                ippmReportSetupDefinition {
```

```
                       onMeasureCycle(2),
                       reportOnlyUptoDownMetricResults(4),
```

```
                        inInformRequestPDU(10),
                        clearHistory(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 from the
   ippmHistoryTable on reception of the response to the InformRequestPDU
   :

                ippmReportSetupDefinition {
                        onMeasureCycle(2),
                        inInformRequestPDU(10),
                        clearHistory(13)
                }
        "
        DEFVAL { { onMeasureCycle, inInformRequestPDU, clearHistory } }
        ::= { ippmReportSetupEntry 1 }

   ippmReportSetupMetricThreshold OBJECT-TYPE
        SYNTAX Integer32
        MAX-ACCESS read-only
        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 }

   ippmReportSetupEventsDurationThreshold OBJECT-TYPE
        SYNTAX Integer32
        UNITS      "Seconds"
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
        "An event is generated when the duration of a series of results,
   which are continuously over the metric threshold, cross over the
   duration of the ippmReportSetupEventsDurationThreshold.
        "
        DEFVAL { 15 }
        ::= { ippmReportSetupEntry 3 }

   ippmReportSetupNMS OBJECT-TYPE
        SYNTAX DisplayString
```

```
        MAX-ACCESS read-only
        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.
     "
     ::= { ippmReportSetupEntry 4 }


--
-- 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.
             "

     ::= { ippmReportGroup 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. The measure and the report definition are identified by the
index members ippmMeasureOwner and ippmMeasureIndex.

In the index :

     + ippmMeasureOwner and ippmMeasureIndex identify the
     ippmMeasureEntry and the ippmReportSetupEntry on whose behalf
     this report was created

     + ippmMetricIndex identifies the ippmMetricEntry of the metric
     measured
     + ippmReportTimeMark value is the absolute time when the value
     of the metric was measured.
```

```
     The ippmReportTimeMark value is the absolute time when the
ippmHistoryValue was performed.
     IppmHistoryValue is the value of the metric measured at the time
ippmReportTimeMark.
     "
     INDEX { ippmMeasureOwner, ippmMeasureIndex, ippmMetricIndex,
ippmReportTimeMark }
     ::= { ippmReportTable 1 }

IppmReportEntry ::=
     SEQUENCE {
             ippmReportTimeMark      GMTDateAndTime,
             ippmReportValue         Integer32
     }
ippmReportTimeMark OBJECT-TYPE
     SYNTAX GMTDateAndTime
     MAX-ACCESS read-only
     STATUS     current
     DESCRIPTION
"The instant of the measure of the result."
     ::= { ippmReportEntry 1 }

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

             "The value."
     ::= { ippmReportEntry 2 }


--
-- ippmNotifications
--





ippmSingletonAlarm    NOTIFICATION-TYPE
     OBJECTS      {
             ippmReportSetupDefinition,
             ippmReportSetupMetricThreshold,
             ippmMetricUnit,
             ippmHistoryValue
     }
     STATUS       current
     DESCRIPTION
```

"A notification sent because 2 contiguous results are on opposite sides of the metric threshold value.

```
                The index of the included ippmReportSetupMetricThreshold
                object identifies the ippmMeasureEntry and the
                ippmResultSetupEntry that specified the threshold.

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

   ippmEventsDurationExceededAlarm    NOTIFICATION-TYPE
        OBJECTS      {
                ippmReportSetupDefinition,
                ippmReportSetupEventsDurationThreshold,
                ippmMetricUnit,
                ippmHistoryValue
        }
        STATUS       current
        DESCRIPTION
                "A notification sent when the duration of contiguous
                raising ippmReportSetupMetricThreshold exceeds the
                ippmReportSetupEventsDurationThreshold value. 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 report. The ippmHistoryTimeMark of the index
                identifies the time theses measures results where
                performed.
                "
        ::= { ippmNotifications 2 }


   ippmCycleOfMeasureReport    NOTIFICATION-TYPE
        OBJECTS      {
                ippmReportSetupDefinition,
                ippmMetricUnit,
                ippmHistoryValue
        }
        STATUS       current
        DESCRIPTION
                "A notification sent when a measure cycle 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 ippmHistoryTimeMark of the index
                identifies the time the measures where performed.
                "
   ::= { ippmNotifications 3 }


   ippmCompletedMeasureReport    NOTIFICATION-TYPE
        OBJECTS      {
                ippmReportSetupDefinition,
                ippmMetricUnit,
                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 ippmHistoryTimeMark of the index
                identifies the time the measures where performed.
                "
        ::= { ippmNotifications 4 }



   --
   -- Compliance statements
   --

   ippmCompliance         MODULE-COMPLIANCE
        STATUS             current
        DESCRIPTION
                "The compliance statement for SNMP entities which
                implement the IPPM MIB"
        MODULE -- this module
        MANDATORY-GROUPS { ippmSystemGroup, ippmMeasureGroup,
   ippmNetworkMeasureGroup, ippmHistoryGroup }
      ::= { ippmCompliances 1 }

   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. References

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

[2]Paxson, V., Almes, G., Mahdavi, J. and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, May 1998.

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

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

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

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

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

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

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

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

[11] 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.

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

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

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

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

[16] 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.

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

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

[19] 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.

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

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

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

[23] Waldbusser, S., "Remote Network Monitoring MIB", STD 59, RFC 2819, Lucent Technologies, May 2000

[24] Waldbusser, S., "Remote Network Monitoring Management Information Base Version 2 using SMIv2", RFC 2021, International Network Services, January 1997.

[25] Remote Network Monitoring MIB Protocol Identifier Reference. A. Bierman, C. Bucci, R. Iddon. RFC RFC2895 ,August 2000.

[26] Remote Network Monitoring MIB Protocol Identifier Macros. A. Bierman, C. Bucci, R. Iddon. RFC RFC2896, August 2000.

11. Acknowledgments

A Kerbe.

12. Author's 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

others, and derivative works that comment on or otherwise explain it or assist its implementation may be prepared, copied, published and