Network Working Group   T. Burbridge
Internet-Draft   P. Eardley
Intended status: Standards Track   BT
Expires: September 6, 2015   M. Bagnulo
                                          Universidad Carlos III de Madrid
                                                         J. Schoenwaelder
                                                  Jacobs University Bremen
                                                            March 5, 2015

        Information Model for Large-Scale Measurement Platforms (LMAP)
                draft-ietf-lmap-information-model-04

Abstract

   This Information Model applies to the Measurement Agent within a
   Large-Scale Measurement Platform. As such it outlines the
   information that is (pre-)configured on the MA or exists in
   communications with a Controller or Collector within an LMAP
   framework. The purpose of such an Information Model is to provide a
   protocol and device independent view of the MA that can be
   implemented via one or more Control and Report protocols.

Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   A large-scale measurement platform is a collection of components that
   work in a coordinated fashion to perform measurements from a large
   number of vantage points. The main components of a large-scale
   measurement platform are the Measurement Agents (hereafter MAs), the
   Controller(s) and the Collector(s).

The MAs are the elements actually performing the measurements. The
MAs are controlled by exactly one Controller at a time and the
Collectors gather the results generated by the MAs. In a nutshell,
the normal operation of a large-scale measurement platform starts
with the Controller instructing a set of one or more MAs to perform a
set of one or more Measurement Tasks at a certain point in time. The
MAs execute the instructions from a Controller, and once they have
done so, they report the results of the measurements to one or more
Collectors. The overall framework for a Large Measurement platform
as used in this document is described in detail in
[I-D.ietf-lmap-framework].

A large-scale measurement platform involves basically three types of
protocols, namely, a Control protocol (or protocols) between a
Controller and the MAs, a Report protocol (or protocols) between the
MAs and the Collector(s) and several measurement protocols between
the MAs and Measurement Peers (MPs), used to actually perform the
measurements. In addition some information is required to be
configured on the MA prior to any communication with a Controller.

This document defines the information model for both Control and the
Report protocols along with pre-configuration information that is
required on the MA before communicating with the Controller, broadly
named as the LMAP Information Model. The measurement protocols are
out of the scope of this document.

As defined in [RFC3444], the LMAP Information Model (henceforth also
referred to as LMAP IM) defines the concepts involved in a large-
scale measurement platform at a high level of abstraction,
independent of any specific implementation or actual protocol used to
exchange the information. It is expected that the proposed
information model can be used with different protocols in different
measurement platform architectures and across different types of MA
devices (e.g., home gateway, smartphone, PC, router).

The definition of an Information Model serves a number of purposes:

1. To guide the standardisation of one or more Control and Report
    protocols and data models

2. To enable high-level inter-operability between different Control
    and Report protocols by facilitating translation between their
    respective data models such that a Controller could instruct sub-
    populations of MAs using different protocols

3. To form agreement of what information needs to be held by an MA
    and passed over the Control and Report interfaces and support the
    functionality described in the LMAP framework

4. Enable existing protocols and data models to be assessed for
    their suitability as part of a large-scale measurement system

[2](#). Notation

This document use an object-oriented programming-like notation to
define the parameters (names/values) of the objects of the
information model. An optional field is enclosed by [ ], and an
array is indicated by two numbers in angle brackets, <m..n>, where m
indicates the minimal number of values, and n is the maximum. The
symbol * for n means no upper bound.

[3](#). LMAP Information Model

The information described herein relates to the information stored,
received or transmitted by a Measurement Agent as described within
the LMAP framework [I-D.ietf-lmap-framework]. As such, some subsets
of this information model are applicable to the measurement
Controller, Collector and any device management system that pre-
configures the Measurement Agent. The information described in these
models will be transmitted by protocols using interfaces between the
Measurement Agent and such systems according to a Data Model.

For clarity the information model is divided into six sections:

1. Pre-Configuration Information. Information pre-configured on the
    Measurement Agent prior to any communication with other
    components of the LMAP architecture (i.e., the Controller,
    Collector and Measurement Peers), specifically detailing how to
    communicate with a Controller and whether the device is enabled
    to participate as an MA.

2. Configuration Information. Update of the pre-configuration
    information during the registration of the MA or subsequent
    communication with the Controller, along with the configuration
    of further parameters about the MA (rather than the Tasks it
    should perform) that were not mandatory for the initial
    communication between the MA and a Controller.

3. Instruction Information. Information that is received by the MA
    from the Controller pertaining to the Tasks that should be
    executed. This includes the task execution Schedules (other than
    the Controller communication Schedule supplied as
    (pre)configuration information) and related information such as
    the Task Configuration, communication Channels to Collectors and
    schedule Timing information. It also includes Task Suppression
    information that is used to over-ride normal Task execution.

4. Logging Information. Information transmitted from the MA to the
    Controller detailing the results of any configuration operations
    along with error and status information from the operation of the
    MA.

5. Capability and Status Information. Information on the general
    status and capabilities of the MA. For example, the set of
    measurements that are supported on the device.

6. Reporting Information. Information transmitted from the MA to
    one or more Collectors including measurement results and the
    context in which they were conducted.

In addition the MA may hold further information not described herein,
and which may be optionally transferred to or from other systems
including the Controller and Collector. One example of information
in this category is subscriber or line information that may be
extracted by a task and reported by the MA in the reporting
communication to a Collector.

It should also be noted that the MA may be in communication with
other management systems which may be responsible for configuring and
retrieving information from the MA device. Such systems, where
available, can perform an important role in transferring the pre-
configuration information to the MA or enabling/disabling the
measurement functionality of the MA.

The Information Model is divided into sub-sections for a number of
reasons. Firstly the grouping of information facilitates reader
understanding. Secondly, the particular groupings chosen are
expected to map to different protocols or different transmissions
within those protocols.

The granularity of data transmitted in each operation of the Control
and Report Protocols is not dictated by the Information Model. For
example, the Instruction object may be delivered in a single
operation. Alternatively, Schedules and Task Configurations may be
separated or even each Schedule/Task Configuration may be delivered
individually. Similarly the Information Model does not dictate
whether data is read, write, or read/write. For example, some
Control Protocols may have the ability to read back Configuration and
Instruction information which have been previously set on the MA.
Lastly, while some protocols may simply overwrite information (for
example refreshing the entire Instruction Information), other
protocols may have the ability to update or delete selected items of
information.

The information in these six sections is captured by a number of
common information objects. These objects are also described later
in this document and comprise of:

1. Schedules. A set of Schedules tell the MA to do something.
    Without a Schedule no Task (from a measurement to reporting or
    communicating with the Controller) is ever executed. Schedules
    are used within the Instruction to specify what tasks should be
    performed, when, and how to direct their results. A Schedule is
    also used within the pre-Configuration and Configuration
    information in order to execute the Task or Tasks required to
    communicate with the Controller.

2. Channels. A set of Channel objects are used to communicate with
    a number of endpoints (i.e. the Controller and Collectors). Each
    Channel object contains the information required for the
    communication with a single endpoint such as the target location
    and security details.

3. Task Configurations. A set of Task Configurations is used to
    configure the Tasks that are run by the MA. This includes the
    registry entry for the Task and any configuration parameters.
    Task Configurations are referenced from a Schedule in order to
    specify what Tasks the MA should execute.

4. Timings. A set of Timing objects that can be referenced from the
    Schedules. Each Schedule always references exactly one Timing
    object. A Timing object specfies either a singleton or series of
    time events. They are used to indicate when Tasks should be
    executed.

The following diagram illustrates the structure in which these common
information objects are referenced. The references are achieved by
each object (Task Configuration, Timing) being given a short text
name that is used by other objects. The objects shown in parenthesis
are part of the internal object structure of a Schedule. Channels
are not shown in the diagram since they are only used as an option by
selected Task Configurations but are similarly referenced using a
short text name.

```
    Schedule
        |----------> Timing
        |----------> (Scheduled Tasks)
                              |----------> Task Configuration
                              |----------> Destination Tasks
```

It should be clear that the top-level bahaviour of an MA is simply to
execute Schedules. Every action referenced by a Schedule is defined

as a Task. As such, these actions are configured through Task
Configurations and executed according to the Timing referenced by the
Schedule in which they appear. Tasks can implement a variety of
different types of actions. While in terms of the Information Model,
all Tasks have the same structure, it can help conceptually to think
of different Task categories:

1. Measurement Tasks measure some aspect of network performance or
    traffic. They may also capture contextual information from the
    MA device or network interfaces such as the device type or
    interface speed.

2. Data Transfer Tasks

    A. Reporting Tasks report the results of Measurement Tasks to
        Collectors

    B. Control Task(s) implement the Control Protocol and
        communicate with the Controller. Depending on the Control
        Protocol there may be a number of specialist tasks such as:
        Configuration Task; Instruction Task; Suppression Task;
        Capabilities Task; Logging Task etc.

3. Data Analysis Tasks can exist to analyse data from other
    Measurement Tasks locally on the MA

4. Data Management Tasks may exist to clean-up, filter or compress
    data on the MA such as Measurement Task results

3.1. Pre-Configuration Information

This information is the minimal information that needs to be pre-
configured to the MA in order for it to successfully communicate with
a Controller during the registration process. Some of the Pre-
Configuration Information elements are repeated in the Configuration
Information in order to allow an LMAP Controller to update these
items. The pre-configuration information also contains some elements
that are not under the control of the LMAP framework (such as the
device identifier and device security credentials).

This Pre-Configuration Information needs to include a URL of the
initial Controller from where configuration information can be
communicated along with the security information required for the
communication including the certificate of the Controller (or the
certificate of the Certification Authority which was used to issue
the certificate for the Controller). All this is expressed as a
Channel. While multiple Channels may be provided in the Pre-

Configuration Information they must all be associated with a single
Controller (e.g. over different interfaces or network protocols).

Where the MA pulls information from the Controller, the Pre-
Configuration Information also needs to contain the timing of the
communication with the Controller as well as the nature of the
communication itself (such as the protocol and data to be
transferred). The timing is given as a Schedule that executes the
Task(s) responsible for communication with the Controller. It is
this Task (or Tasks) that implement the Control protocol between the
MA and the Controller and utilises the Channel information. The
Task(s) may take additional parameters in which case a Task
Configuration can also be included.

Even where information is pushed to the MA from the Controller
(rather than pulled by the MA), a Schedule still needs to be
supplied. In this case the Schedule will simply execute a Controller
listener task when the MA is started. A Channel is still required
for the MA to establish secure communication with the Controller.

It can be seen that these Channels, Schedules and Task Configurations
for the initial MA-Controller communication are no different in terms
of the Information Model to any other Channel, Schedule or Task
Configuration that might execute a Measurement Task or report the
measurement results (as described later).

The MA may be pre-configured with an MA ID, or may use a Device ID in
the first Controller contact before it is assigned an MA ID. The
Device ID may be a MAC address or some other device identifier
expressed as a URN. If the MA ID is not provided at this stage then
it must be provided by the Controller during Configuration.

Detail of the information model elements:

```
// MA pre-configuration minimal information to communicate
// initially with Controller

object {
    [uuid               ma-agent-id;]
     ma-task-obj        ma-control-tasks<1..*>;
     ma-channel-obj     ma-control-channels<1..*>;
     ma-schedule-obj    ma-control-schedules<1..*>;
    [urn                ma-device-id;]
     credentials        ma-credentials;
} ma-config-obj;
```

The details of the Channel and Schedule objects are described later
since they are common to several parts of the information model.

3.2. Configuration Information

   During registration or at any later point at which the MA contacts
   the Controller (or vice-versa), the choice of Controller, details for
   the timing of communication with the Controller or parameters for the
   communication Task(s) can be changed (as captured by the Channels,
   Schedules and Task Configurations objects). For example the pre-
   configured Controller (specified as a Channel or Channels) may be
   over-ridden with a specific Controller that is more appropriate to
   the MA device type, location or characteristics of the network (e.g.
   access technology type or broadband product). The initial
   communication Schedule may be over-ridden with one more relevant to
   routine communications between the MA and the Controller.

   While some Control protocols may only use a single Schedule, other
   protocols may use several Schedules (and related data transfer Tasks)
   to update the Configuration Information, transfer the Instruction
   Information, transfer Capability and Status Information and send
   other information to the Controller such as log or error
   notifications. Multiple Channels may be used to communicate with the
   same Controller over multiple interfaces (e.g. to send logging
   information over a different network).

   In addition the MA will be given further items of information that
   relate specifically to the MA rather than the measurements it is to
   conduct or how to report results. The assignment of an ID to the MA
   is mandatory. If the MA Agent ID was not optionally provided during
   the pre-configuration then one must be provided by the Controller
   during Configuration. Optionally a Group ID may also be given which
   identifies a group of interest to which that MA belongs. For example
   the group could represent an ISP, broadband product, technology,
   market classification, geographic region, or a combination of
   multiple such characteristics. Where the Measurement Group ID is set
   an additional flag (the Report MA ID flag) is required to control
   whether the Measurement Agent ID is also to be reported. The
   reporting of a Group ID without the MA ID allows the MA to remain
   anonymous, which may be particularly useful to prevent tracking of
   mobile MA devices.

   Optionally an MA can also be configured to stop executing any
   Instruction Schedule if the Controller is unreachable. This can be
   used as a fail-safe to stop Measurement and other Tasks being
   conducted when there is doubt that the Instruction Information is
   still valid. This is simply represented as a time window in
   milliseconds since the last communication with the Controller after
   which Instruction Schedules are to be suspended. The appropriate
   value of the time window will depend on the specified communication
   Schedule with the Controller and the duration for which the system is

willing to tolerate continued operation with potentially stale
Instruction Information.

While Pre-Configuration Information is persistent upon device reset
or power cycle, the persistency of the Configuration Information may
be device dependent. Some devices may revert back to their pre-
configuration state upon reboot or factory reset, while other devices
may store all Configuration and Instruction information in persistent
storage. A Controller can check whether an MA has the latest
Configuration and Instruction information by examining the Capability
and Status information for the MA.

It should be noted that control shcedules and tasks cannot be
suppressed as evidenced by the lack of suppression information in the
Configuration. The control schedule must only reference tasks listed
as control tasks (i.e. within the Configuration information). Any
suppress-by-default flag against control tasks will be ignored.

Detail of the additional and updated information model elements:

// MA Configuration

```
object {
    uuid                ma-agent-id;
    ma-task-obj         ma-control-tasks<1..*>;
    ma-channel-obj      ma-control-channels<1..*>;
    ma-schedule-obj     ma-control-schedules<1..*>;
    [urn                ma-device-id;]
     credentials        ma-credentials;
    [string             ma-group-id;]
    [boolean            ma-report-ma-id-flag;]
    [int                ma-control-channel-failure-threshold;]
} ma-config-obj;
```

3.3. Instruction Information

The Instruction information model has four sub-elements:

1. Instruction Task Configurations

2. Report Channels

3. Instruction Schedules

4. Suppression

The Instruction supports the execution of all Tasks on the MA except
those that deal with communication with the Controller (specified in

(pre-)configuration information). The Tasks are configured in
Instruction Task Configurations and included by reference in
Instruction Schedules that specify when to execute them. The results
can be communicated to other Tasks or a Task may implement a
Reporting Protocol and communicate results over Report Channels.
Suppression is used to temporarily stop the execution of new Tasks as
specified by the Instruction Schedules (and optionally to stop
ongoing Tasks).

A Task Configuration is used to configure the mandatory and optional
parameters of a Task. It also serves to instruct the MA about the
Task including the ability to resolve the Task to an executable and
specifying the schema for the Task parameters.

A Report Channel defines how to communicate with a single remote
system specified by a URL. A Report Channel is used to send results
to single Collector but is no different in terms of the Information
Model to the Control Channel used to transfer information between the
MA and the Controller. Several Report Channels can be defined to
enable results to be split or duplicated across different
destinations. A single Channel can be used by multiple (reporting)
Task Configurations to transfer data to the same Collector. A single
Reporting Task Configuration can also be included in multiple
Schedules. E.g. a single Collector may receive data at three
different cycle rates, one Schedule reporting hourly, another
reporting daily and a third specifying that results should be sent
immediately for on-demand measurement tasks. Alternatively multiple
Report Channels can be used to send Measurement Task results to
different Collectors. The details of the Channel element is
described later as it is common to several objects.

Instruction Schedules specify which Tasks to execute according to a
given Timing (that can execute a single or repeated series of Tasks).
The Schedule also specifies how to link Tasks output data to other
scheduled Tasks - i.e. sending selected outputs to other Tasks.

Measurement Suppression information is used to over-ride the
Instruction Schedule and temporarily stop measurements or other Tasks
from running on the MA for a defined or indefinite period. While
conceptually measurements can be stopped by simply removing them from
the Measurement Schedule, splitting out separate information on
Measurement Suppression allows this information to be updated on the
MA on a different timing cycle or protocol implementation to the
Measurement Schedule. It is also considered that it will be easier
for a human operator to implement a temporary explicit suppression
rather than having to move to a reduced Schedule and then roll-back
at a later time.

The explicit Suppression instruction message is able to simply
enable/disable all Instruction Tasks (that are enabled for default
suppression) as well as having fine control on which Tasks are
suppressed. Suppression of both specified Task Configurations and
Measurement Schedules is supported. Support for disabling specific
Task Configurations allows malfunctioning or mis-configured Tasks or
Task Configurations that have an impact on a particular part of the
network infrastructure (e.g., a particular Measurement Peer) to be
targeted. Support for disabling specific Schedules allows for
particularly heavy cycles or sets of less essential Measurement Tasks
to be suppressed quickly and effectively. Note that Suppression has
no effect on either Controller Tasks or Controller Schedules.

When no tasks or schedules are explicitly listed, all Instruction
tasks will be suppressed (or not) as indicated by the suppress-by-
default flag in the Task Configuration. If tasks or schedules are
listed explicitly then only these listed tasks or schedules will be
suppressed regardless of the suppress-by-default flag. If both
individual tasks and individual schedules are listed then only the
listed schedules, plus the listed tasks where present in other
schedules, will be suppressed regardless of the suppress-by-default
flag.

Suppression stops new Tasks from executing. In addition, the
Suppression information also supports an additional Boolean that is
used to select whether on-going tasks are also to be terminated.

Unsuppression is achieved through either overwriting the Measurement
Suppression information (e.g. changing 'enabled' to False) or through
the use of an End time such that the Measurement Suppression will no
longer be in effect beyond this time. The datetime format used for
all elements in the information model (e.g. the suppression start and
end dates) MUST conform to RFC 3339 [RFC3339].

The goal when defining these four different elements is to allow each
part of the information model to change without affecting the other
three elements. For example it is envisaged that the Report Channels
and the set of Task Configurations will be relatively static. The
Instruction Schedule, on the other hand, is likely to be more
dynamic, as the measurement panel and test frequency are changed for
various business goals. Another example is that measurements can be
suppressed with a Suppression command without removing the existing
Instruction Schedules that would continue to apply after the
Suppression expires or is removed. In terms of the Controller-MA
communication this can reduce the data overhead. It also encourages
the re-use of the same standard Task Configurations and Reporting
Channels to help ensure consistency and reduce errors.

Definition of the information model elements:

```
// Instruction to the MA to configure Tasks, Channels,
//Schedules and Suppression

object {
    ma-task-obj         ma-instruction-tasks<0..*>;
    ma-channel-obj      ma-report-channels<0..*>;
    ma-schedule-obj     ma-instruction-schedules<0..*>;
    ma-suppression-obj  ma-suppression;
} ma-instruction-obj;

// Suppression object to temporarily override new task execution
// in Instructions and optionally stop currently running tasks

object {
    boolean             ma-suppression-enabled;
    [boolean            ma-suppression-stop-ongoing-tasks;]
                // default: false
    [datetime           ma-suppression-start;] // default: immediate
    [datetime           ma-suppression-end;]   // default: indefinite
    [string             ma-suppression-task-names<0..*>;]
                        // default: all tasks if
                        // ma-suppression-task-names is empty
    [string             ma-suppression-schedule-names<0..*>;]
                        // default: all schedules if
                        // ma-suppression-schedule-names is empty
} ma-suppression-obj;
```

3.4. Logging Information

The MA may report on the success or failure of Configuration or
Instruction communications from the Controller. In addition further
operational logs may be produced during the operation of the MA and
updates to capabilities may also be reported. Reporting this
information is achieved in exactly the same manner as scheduling any
other Task. We make no distinction between a Measurement Task
conducting an active or passive network measurement and one which
solely retrieves static or dynamic information from the MA such as
capabilities or logging information. One or more logging tasks can
be programmed or configured to capture subsets of the Logging
Information. These logging tasks are then executed by Schedules
which also specify that the resultant data is to be transferred over
the Controller Channels.

The type of Logging Information will fall into three different
categories:

1. Success/failure/warning messages in response to information
    updates from the Controller. Failure messages could be produced
    due to some inability to receive or parse the Controller
    communication, or if the MA is not able to act as instructed.
    For example:

    * "Measurement Schedules updated OK"

    * "Unable to parse JSON"

    * "Missing mandatory element: Measurement Timing"

    * "'Start' does not conform to schema - expected datetime"

    * "Date specified is in the past"

    * "'Hour' must be in the range 1..24"

    * "Schedule A refers to non-existent Measurement Task
       Configuration"

    * "Measurement Task Configuration X registry entry Y not found"

    * "Updated Measurement Task Configurations do not include M used
       by Measurement Schedule N"

2. Operational updates from the MA. For example:

    * "Out of memory: cannot record result"

    * "Collector 'collector.example.com' not responding"

    * "Unexpected restart"

    * "Suppression timeout"

    * "Failed to execute Measurement Task Configuration H"

3. Status updates from the MA. For example:

    * "Device interface added: eth3 "

    * "Supported measurements updated"

    * "New IP address on eth0: xxx.xxx.xxx.xxx"

This Information Model document does not detail the precise format of
logging information since it is to a large extent protocol and MA
specific. However, some common information can be identified.

MA Logging information model elements:

// Logging object

```
object {
    uuid               ma-log-agent-id;
    datetime           ma-log-event-time;
    code               ma-log-code;
    string             ma-log-description;
} ma-log-obj;
```

3.5. Capability and Status Information

The MA will hold Capability Information that can be retrieved by a
Controller. Capabilities include the device interface details
available to Measurement Tasks as well as the set of Measurement
Tasks/Roles (specified by a registry entry) that are actually
installed or available on the MA. Status information includes the
times that operations were last performed such as contacting the
Controller or producing Reports.

MA Status information model elements:

// Main MA Status information object

```
object {
    uuid                    ma-agent-id;
    urn                     ma-device-id;
    string                  ma-hardware;
    string                  ma-firmware;
    string                  ma-version;
    ma-interface-obj        ma-interfaces<0..*>;
    ma-task-capability-obj  ma-supported-tasks<0..*>;
    datetime                ma-last-started;
   [ma-condition-obj        ma-conditions<0..*>;]
   [ma-task-status-obj      ma-task-status<0..*>;]
} ma-status-obj;
```

```
    // Per-Task status information and conditions

    object {
        string              ma-task-name;
        string              ma-task-role;
        uri                 ma-task-registry;
        datetime            ma-task-last-invocation;
        datetime            ma-task-last-successful;
        string              ma-task-last-successful-message;
        datetime            ma-task-last-failed;
        string              ma-task-last-failed-message;
       [ma-condition-obj    ma-task-conditions<0..*>];
    } ma-task-status-obj


    // Additional status conditions

    object {
      int                   ma-condition-code;
      string                ma-condition-text;
    } ma-condition-obj


    // Interface information

    object {
        string              ma-interface-name;
        string              ma-interface-type;
       [int                 ma-interface-speed;]  // bps
       [string              ma-link-layer-address;]
       [ip-address          ma-interface-ip-addresses<0..*>];
       [ip-address          ma-interface-gateways<0..*>;]
       [ip-address          ma-interface-dns-servers<0..*>;]
    } ma-interface-obj;

    // Supported tasks/roles

    object {
        string              ma-task-name;
        string              ma-task-role;
        uri                 ma-task-registry;
    } ma-task-capability-obj;
```

3.6. Reporting Information

   At a point in time specified by a Schedule, the MA will execute a
   task or tasks that communicate a set of measurement results to the

Collector. These Reporting Tasks will be configured to transmit task
results over a specified Report Channel to a Collector.

It should be noted that the output from Tasks does not need to be
sent to communication Channels. It can alternatively, or
additionally, be sent to other Tasks on the MA. This facilitates
using a first Measurement Task to control the operation of a later
Measurement Task (such as first probing available line speed and then
adjusting the operation of a video testing measurement) and also to
allow local processing of data to output alarms (e.g. when
performance drops from earlier levels). Of course, subsequent Tasks
also include Tasks that implement the reporting protocol(s) and
transfer data to one or more Collector(s).

The Report generated by a Reporting Task is structured hierarchically
to avoid repetition of report header and Measurement Task
Configuration information. The report starts with the timestamp of
the report generation on the MA and details about the MA including
the optional Measurement Agent ID and Group ID (controlled by the
Configuration Information).

Much of the report Information is optional and will depend on the
implementation of the Reporting Task and any parameters defined in
the Task Configuration for the Reporting Task. For example some
Reporting Tasks may choose not to include the Measurement Task
Configuration or scheduled task parameters, while others may do so
dependent on the Controller setting a configurable parameter in the
Task Configuration.

It is possible for a Reporting Task to send just the Report header
(datetime and optional agent ID and/or Group ID) if no measurement
data is available. Whether to send such empty reports again is
dependent on the implementation of the Reporting Task and potential
Task Configuration parameter.

The handling of measurement data on the MA before generating a Report
and transfer from the MA to the Collector is dependent on the
implementation of the device, MA and/or scheduled Tasks and not
defined by the LMAP standards. Such decisions may include limits to
the measurement data storage and what to do when such available
storage becomes depleted.

No context information, such as line speed or broadband product are
included within the report header information as this data is
reported by individual tasks at the time they execute. Either a
Measurement Task can report contextual parameters that are relevant
to that particular measurement, or specific tasks can be used to

gather a set of contextual and environmental data. at certain times independent of the reporting schedule.

After the report header information the results are reported grouped according to different Measurement Task Configurations. Each Task section optionally starts with replicating the Measurement Task Configuration information before the result headers (titles for data columns) and the result data rows. The Options reported are those used for the scheduled execution of the Measurement Task and therefore include the Options specified in the Task Configuration as well as additional Options specified in the Scheduled Task. The Scheduled Task Options are appended to the Task Configuration Options in exactly the same order as they were provided to the Task during execution.

The result row data includes a time for the start of the measurement and optionally an end time where the duration also needs to be considered in the data analysis.

Some Measurement Tasks may optionally include an indication of the cross-traffic although the meaning a definition of cross-traffic is left up to each individual Measurement Task. Some Measurement Tasks may also output other environmental measures in addition to cross-traffic such as CPU utlilisation or interface speed.

Where the Configuration and Instruction information represent information transmitted via the Control Protocol, the Report represents the information that is transmitted via the Report Protocol. It is constructed at the time of sending a report and represents the inherent structure of the information that is sent to the Collector.

Information model elements:

// Main Report object with report header information

```
object {
    datetime            ma-report-date;
   [uuid                ma-report-agent-id;]
   [string              ma-report-group-id;]
   [ma-report-task-obj  ma-report-tasks<0..*>];
} ma-report-obj;
```

```
   // Report task header information

   object {
       string              ma-report-task-name;
     [uri                  ma-report-task-registry-entry;]
     [name-value-pair      ma-report-scheduled-task-options<0..*>];
     [string               ma-report-task-cycle-id;]
       string              ma-report-task-column-labels<0..*>;
       ma-result-row-obj   ma-report-task-rows<0..*>;
   } ma-report-task-obj;

   // Report tasks result rows

   object {
       datetime            ma-report-result-start-time;
     [datetime             ma-report-result-end-time;]
       string              ma-report-result-conflicting-tasks<0..*>;
       data                ma-report-result-values<0..*>;
   } ma-result-row-obj;
```

3.7. Common Objects

3.7.1. Schedules

   A Schedule specifies the execution of a single or repeated series of
   Tasks. Each Schedule contains basically two elements: a list of
   Tasks to be executed and a timing object for the Schedule. The
   Schedule states what Tasks to run (with what configuration) and when
   to run the Tasks.

   Multiple Tasks in the list of a single Measurement Schedule will be
   executed in order with minimal gaps. Tasks in different Schedules
   execute in parallel with such conflicts being reported in the
   Reporting Information. If two or more Schedules have the same start
   time, then the two will execute in parallel. There is no mechanism
   to prioritise one schedule over another or to mutex scheduled tasks.

   As well as specifying which Tasks to execute, the Schedule also
   specifies how to link the data outputs from each scheduled task to
   other scheduled tasks. Specifying this within the Schedule allows
   the highest level of flexibility since it is even possible to send
   the output from different executions of the same Task Configuration
   to different destinations. Since a single Task may have multiple
   outputs, the Schedule can independently specify which outputs go to
   which destinations. For example, a Measurement Task might report
   routine results to a data Reporting Task that communicates hourly via
   the Broadband PPP interface, but also outputs emergency conditions
   via an alarm Reporting Task communicating immediately over a GPRS

channel. Note that task-to-task data transfer is always specified in
association with the scheduled execution of the sending task - there
is no need for a corresponding input specification for the receiving
task. While it is likely that an MA implementation will use a queue
mechanism between the scheduled tasks, this Information Model does
not mandate or define a queue, or any potential associated parameters
such as storage size and retention policies.

When specifying the task to execute within the Schedule, it is
possible to add to the task configuration option parameters. This
allows the Task Configuration to determine the common characteristics
of a Task, while selected parameters (e.g. the test target URL) are
defined within the schedule. A single Tasks Configuration can even
be used multiple times in the same schedule with different additional
parameters. This allows for efficiency in creating and transferring
the Instruction. Note that the semantics of what happens if an
option is defined multiple times (either in the Task Configuration,
Schedule or in both) is not standardised and will depend upon the
Task. For example, some tasks may legitimately take multiple values
for a single parameter.

Where Options are specified in both the Schedule and the Task
Configuration, the Schedule Options are appended to those specified
in the Task Configuration.

Example: A Schedule references a single Measurement Task
   Configuration for the UDP latency. It specifies that results are
   to be sent to a scheduled Reporting Task. This Reporting Task is
   executed by a separate Schedule that specifies that it should run
   hourly at 5 minutes past the hour. When run this Reporting Task
   takes the data generated by the UDP latency Task as well as any
   other data to be included in the hourly report and transfers it to
   the Collector over the Report Channel specified within its own
   Schedule.

// main Schedule object with Timing and list of Scheduled Tasks

```
object {
    string              ma-schedule-name;
    ma-sched-task-obj   ma-schedule-tasks<0..*>;
    ma-timing-obj       ma-schedule-timing;
} ma-schedule-obj;
```

```
// Scheduled Task object with reference (by name string) to Task
// Configuration and mappings of data outputs to destination tasks

object {
    string                          ma-schedule-task-name;
    [name-value-pair                ma-schedule-task-options<0..*>];
    [ma-sched-downstream-tasks-obj  ma-schedule-destination-tasks<0..*>;]
 } ma-sched-task-obj;

// Specification of destination scheduled tasks using reference
// to schedule and task configuration names. Mapping of
// integer denoted data outputs to destination scheduled task

object {
    [string    ma-schedule-task-destination-schedule-name];
    [string    ma-schedule-task-destination-task-configuration-name];
    [int       ma-schedule-task-output-selection<0..*>;]  // default: all
} ma-sched-destination-tasks-obj;
```

   Example: A measurement task has two defined inter-task outputs, one
      for routine measurement results and one for errors during the task
      execution. These are defined as available outputs by the task and
      are denoted by the integers 1 & 2. In this example, both outputs
      are sent to the same reporting task called "Hourly reporting Task"
      that is executed from the "Hourly Schedule" schedule. This is
      done by creating a ma-sched-destination-tasks-obj with the output
      selection as [1,2] and the destination task configuration name as
      ["Hourly Reporting Task"] and the destination schedule name as
      "Hourly Schedule".

```
      Measurement Task
        Output 1 -----+----> "Hourly Schedule":"Hourly Reporting Task"
        Output 2 ----/
```

3.7.2. Channels

   A Channel defines a bi-directional communication channel between the
   MA and a Controller or Collector. Multiple Channels can be defined
   to enable results to be split or duplicated across different
   Collectors.

   Each Channel contains the details of the remote endpoint (including
   location and security credential information such as the
   certificate). The timing of when to communicate over a Channel is
   specified by the Schedule which executes the corresponding Control or
   Reporting Task. The certificate can be the digital certificate
   associated to the FQDN in the URL or it can be the certificate of the

Certification Authority that was used to issue the certificate for
the FQDN (Fully Qualified Domain Name) of the target URL (which will
be retrieved later on using a communication protocol such as TLS).
In order to establish a secure channel, the MA will use it's own
security credentials (in the Configuration Information) and the given
credentials for the individual Channel end-point.

As with the Task Configurations, each Channel is also given a text
name by which it can be referenced as a Task Option.

Although the same in terms of information, Channels used for
communication with the Controller are referred to as Control Channels
whereas Channels to Collectors are referred to as Report Channels.
Hence Control Channels will be referenced from Control Tasks executed
by a Control Schedule, whereas Report Channels will be referenced
from within Reporting Tasks executed by an Instruction Schedule.

Multiple interfaces are also supported. For example the Reporting
Task could be configured to send some results over GPRS. This is
especially useful when such results indicate the loss of connectivity
on a different network interface.

Example: A Channel using for reporting results may specify that
   results are to be sent to the URL (https://collector.foo.org/
   report/), using the appropriate digital certificate to establish a
   secure channel..

// Channel object with name string allowing reference.
// Contains channel endpoint target URL and security credentials
// to establish secure channel. Optionally allows interface
// specification (by interface name string reference)

object {
    string   ma-channel-name;
    url   ma-channel-target;
    credentials   ma-channel-credentials;
    [string   ma-channel-interface-name;]
} ma-channel-obj;

3.7.3. Task Configurations

Conceptually each Task Configuration defines the parameters of a Task
that the Measurement Agent (MA) may perform at some point in time.
It does not by itself actually instruct the MA to perform them at any
particular time (this is done by a Schedule). Tasks can be
Measurement Tasks (i.e. those Tasks actually performing some type of
passive or active measurement) or any other scheduled activity
performed by the MA such as transferring information to or from the

Controller and Collectors. Other examples of Tasks may include data
manipulation or processing Tasks conducted on the MA.

A Measurement Task Configuration is the same in information terms to
any other Task Configuration. Both measurement and non-measurement
Tasks have a registry entry to enable the MA to uniquely identify the
Task it should execute and retrieve the schema for any parameters
that may be passed to the Task. This registry entry is specified as
a URI and can therefore be used to identify the Task within a
namespace or point to a web or local file location for the Task
information. As mentioned previously this entry may be used to
identify the Measurement Task in a public namespace
[I-D.ietf-ippm-metric-registry] .

Example: A Measurement Task Configuration may configure a single
   Measurement Task for measuring UDP latency. The Measurement Task
   Configuration could define the destination port and address for
   the measurement as well as the duration, internal packet timing
   strategy and other parameters (for example a stream for one hour
   and sending one packet every 500 ms). It may also define the
   output type and possible parameters (for example the output type
   can be the 95th percentile mean) where the measurement task
   accepts such parameters. It does not define when the task starts
   (this is defined by the Schedule element), so it does not by
   itself instruct the MA to actually perform this Measurement Task.

The Task Configuration will include a local short name for reference
by a Schedule. Task Configurations will also contain a registry
entry as described above. In addition the Task can be configured
through a set of configuration Options. The nature and number of
these Options will depend upon the Task. These options are expressed
as name-value pairs although the 'value' may be a structured object
instead of a simple string or numeric value. The implementation of
these name-value pairs will vary between data models such as JSON,
XML or TR-069.

A Option that must be present for Reporting Tasks is the Channel
reference specifying how to communicate with a Collector. This is
included in the task options and will have a value that matches a
channel name that has been defined in the Instruction. Similarly
Control Tasks will have a similar option with the value set to a
specified Control Channel.

A reporting task might also have a flag parameter to indicate whether
to report if there is no measurement result data pending to be
transferred to the Collector. In addition many tasks will also take
as a parameter which interface to operate over.

   The Task Configuration also contains a suppress-by-default flag that
   specifies the behaviour of a default suppress instruction (that does
   not list explicit tasks or schedules). If this flag is set to FALSE
   then the Task will not be suppressed. It should be noted that
   Controller Tasks are not subject to the suppression instruction and
   therefore this flag will be ignored in such cases.

   In addition the Task Configuration may optionally also be given a
   Measurement Cycle ID. The purpose of this ID is to easily identify a
   set of measurement results that have been produced by Measurement
   Tasks with comparable Options. This ID could be manually incremented
   or otherwise changed when an Option change is implemented which could
   mean that two sets of results should not be directly compared.

```
 // Task Configuration object with string name to allow reference
 // from Schedule. Contains URI to link to registry or local
 // specification of the Task. Options allow the configuration
 // of Task parameters (in the form of name-value pairs)

 object {
     string              ma-task-name;
     uri                 ma-task-registry-entry;
    [ma-task-option      ma-task-options<0..*>];
    [boolean             ma-task-suppress-by-default;] // default: TRUE
    [string              ma-task-cycle-id;]
 } ma-task-obj;
```

   While many of the Task Configuration Options are left to individual
   tasks to define, some common Options are used by multiple tasks and
   benefit from standardisation. These Options are Channel and Role.

   Channel is used to specify the details of an endpoint for Control or
   Reporting Task communications and is detailed elsewhere in this
   document.

   Role is used to specify which Role the task should be performing (as
   defined in the registry) if multiple roles are available.

```
// General Task Option

object {
    string      ma-option-name;
    object      ma-option-value;
} ma-task-option


// Channel Option

oobject {
    string      ma-option-name;    // set to "channel"
    string      ma-option-value;   // set to ma-channel-name reference
} ma-task-option


// Role Option

object {
    string      ma-option-name;    // set to "role"
    string      ma-option-value;   // set to registry role reference
} ma-task-option
```

[3.7.4](#). **Timing Information**

The Timing information object used throughout the information models
can take one of five different forms:

1. Periodic. Specifies a start, end and interval time in
    milliseconds

2. Calendar: Specifies a calendar based pattern - e.g. 22 minutes
    past each hour of the day on weekdays

3. One Off: A single instance occurring at a specific time

4. Immediate: Should occur as soon as possible

5. Startup: Should occur whenever the MA is started (e.g. at device
    startup)

Optionally each of the options may also specify a randomness that
should be evaluated and applied separately to each indicated event.
This randomness parameter defines a uniform interval in milliseconds
over which the start of the task is delayed from the starting times
specified by the timing object.

Both the Periodic and Calendar timing objects allow for a series of
tasks to be executed. While both have an optional end time, it is
best practice to always configure an end time and refresh the
information periodically to ensure that lost MAs do not continue
their tasks forever.

Starup timing is only executed on device startup - not when a new
Instruction is transferred to the MA. If scheduled task execution is
desired both on the transfer of the Instruction and on device restart
then both the Immediate and Startup timing needs to be used in
conjunction.

The datetime format used for all elements in the information model
MUST conform to RFC 3339 [RFC3339].

```
// Main Timing object with name string to allow reference by Schedule
// Must be specialised by one of the Timing options.
// Includes optional uniform random spread in ms from start time
// given by Timing specialisation

object {
   [string             ma-timing-name;]
   union {
       ma-periodic-obj  ma-timing-periodic;
       ma-calendar-obj  ma-timing-calendar;
       ma-one-off-obj   ma-timing-one-off;
       ma-immediate-obj ma-timing-immediate;
       ma-startup-obj   ma-timing-startup;
   }
   [int                ma-timing-random-spread;] // milliseconds
} ma-timing-obj;
```

3.7.4.1. Periodic Timing

Information model elements:

```
// Timing specialisation to run a series of Tasks repeated at
// set intervals

object {
   [datetime        ma-periodic start;]   // default: immediate
   [datetime        ma-periodic-end;]     // default: indefinite
    int             ma-periodic-interval; // milliseconds
} ma-periodic-obj;
```

3.7.4.2. **Calendar Timing**

   Calendar Timing supports the routine execution of Measurement Tasks
   at specific times and/or on specific dates. It can support more
   flexible timing than Periodic Timing since the Measurement Task
   execution does not have to be uniformly spaced. For example a
   Calendar Timing could support the execution of a Measurement Task
   every hour between 6pm and midnight on weekdays only.

   Calendar Timing is also required to perform measurements at
   meaningful instances in relation to network usage (e.g., at peak
   times). If the optional timezone offset is not supplied then local
   system time is assumed. This is essential in some use cases to
   ensure consistent peak-time measurements as well as supporting MA
   devices that may be in an unknown timezone or roam between different
   timezones (but know their own timezone information such as through
   the mobile network).

   Days of week are define using three character strings "Mon", "Tue",
   "Wed", "Thu", "Fri", "Sat", "Sun".

   If a day of the month is specified that does not exist in the month
   (e.g. 29 in Feburary) then those values are ignored.

   The calendar elements within the Calendar Timing do not have defaults
   in order to avoid accidental high-frequency execution of Tasks. If
   all possible values for an element are desired then the wildcard * is
   used.

   Information model elements:

```
// Timing specialisation to run repeated Tasks at specific
// times and/or days

object {
   [datetime       ma-calendar-start;] // default: immediate
   [datetime       ma-calendar-end;]   // default: indefinite
   [int            ma-calendar-months<0..*>;]   // values: 1-12,*
   [days           ma-calendar-days-of-week<0..*>;]
         // values:  "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun",*
   [int            ma-calendar-days-of-month<0..*>;] // values 1-31,*
   [int            ma-calendar-hours<0..*>;]    // values: 0-23,*
   [int            ma-calendar-minutes<0..*>;]  // values: 0-59,*
   [int            ma-calendar-seconds<0..*>;]  // values: 0-59,*
   [int            ma-calendar-timezone-offset;]
           // default: system timezone offset
} ma-calendar-obj;
```

3.7.4.3. One-Off Timing

Information model elements:

// Timing specialisation to run once at a specified time/date

```
object {
    datetime            ma-one-off-time;
} ma-one-off-obj;
```

3.7.4.4. Immediate Timing

The immediate timing object has no further information elements. The
measurement or report is simply to be done as soon as possible.

// Timing specialisation to run immediately

```
object {

                        // empty
} ma-immediate-obj;
```

3.7.4.5. Startup Timing

The immediate timing object has no further information elements. The
measurement or report is simply done at MA initiation.

// Timing specialisation to run at MA startup

```
object {

                        // empty
} ma-startup-obj;
```

4. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an
RFC.

5. Security Considerations

This Information Model deals with information about the control and
reporting of the Measurement Agent. There are broadly two security
considerations for such an Information Model. Firstly the
Information Model has to be sufficient to establish secure
communication channels to the Controller and Collector such that
other information can be sent and received securely. Additionally,
any mechanisms that the Network Operator or other device

administrator employs to pre-configure the MA must also be secure to
protect unauthorized parties from modifying pre-configuration
information. These mechanisms are important to ensure that the MA
cannot be hijacked, for example to participate in a DDoS attack.

The second consideration is that no mandated information items should
pose a risk to confidentiality or privacy given such secure
communication channels. For this latter reason items such as the MA
context and MA ID are left optional and can be excluded from some
deployments. This would, for example, allow the MA to remain
anonymous and for information about location or other context that
might be used to identify or track the MA to be omitted or blurred.

The Information Model should support wherever relevant, all the
security and privacy requirements associated with the LMAP Framework.

6. Acknowledgements

The notation was inspired by the notation used in the ALTO protocol
specification.

Philip Eardley, Trevor Burbridge, Marcelo Bagnulo and Juergen
Schoenwaelder work in part on the Leone research project, which
receives funding from the European Union Seventh Framework Programme
[FP7/2007-2013] under grant agreement number 317647.

7. References

7.1. Normative References

[I-D.ietf-lmap-framework]
           Eardley, P., Morton, A., Bagnulo, M., Burbridge, T.,
           Aitken, P., and A. Akhter, "A framework for large-scale
           measurement platforms (LMAP)", draft-ietf-lmap-
           framework-10 (work in progress), January 2015.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
           Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the
           Internet: Timestamps", RFC 3339, July 2002.

7.2. Informative References

   [I-D.ietf-ippm-metric-registry]
             Bagnulo, M., Claise, B., Eardley, P., Morton, A., and A.
             Akhter, "Registry for Performance Metrics", draft-ietf-
             ippm-metric-registry-01 (work in progress), September
             2014.

   [I-D.schoenw-lmap-yang]
             Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for
             LMAP Measurement Agents", draft-schoenw-lmap-yang-02 (work
             in progress), January 2015.

   [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between
             Information Models and Data Models", RFC 3444, January
             2003.

Appendix A. JSON Data Model Example

   In order to give an example of data in the Information Model we need
   to select a data model language. In the following example we have
   expressed the Data Model using JSON as this will be of direct
   interest to some Control and Report Protocols. A YANG data model
   implementation of the Information Model is provided in a separate
   draft [I-D.schoenw-lmap-yang].

   The example is broken down into a number of different steps that
   might adhere to the steps within a Control and Report Protocol:

   1. Pre-configuration.

   2. Configuration

   3. Capabilities

   4. Instruction

   5. Report

   6. Suppression

   While the pre-configuration is not delivered as part of the Control
   Protocol, the same JSON data model is used for consistency and to aid
   the reader.

```
//Pre-configuration

{
  "ma-config": {
    "ma-agent-id": "550e8400-e29b-41d4-a716-446655440000",
    "ma-control-tasks": [
      {
        "ma-task-name": "Controller configuration",
        "ma-task-registry-entry":
          "urn:ietf:lmap:control:http_controller_configuration",
        "ma-task-options": [{"name": "channel",
          "value": "Controller channel"}]
      }
    ],
    "ma-control-channels": [
      {
        "ma-channel-name": "Controller channel",
        "ma-channel-target": "http://www.example.com/lmap/controller",
        "ma-channel-credientials": { }
      }
    ],
    "ma-control-schedules": [
      {
        "ma-schedule-name": "pre-configured schedule",
        "ma-schedule-tasks": {

            "ma-schedule-task-name": "Controller configuration",
        },
        "ma-schedule-timing": {
          "ma-timing-name": "startup plus up to one hour",
          "ma-timing-startup": {
          },
          "ma-timing-random-spread": "3600000"
        }
      }
    ],
    "ma-credentials": { }
  }
}
```

Given the pre-configuration information the MA is able to contact the
Controller and receive an updated/expanded Configuration. In this
example additional Control Protocol tasks to post Status and
Capabilities to the Controller and fetch the Instruction are added as

well as moving the schedule timing for contacting the Controller to
hourly.

```
// Configuration

{
  "ma-config": {
    "ma-agent-id": "550e8400-e29b-41d4-a716-446655440000",
    "ma-control-tasks": [
      {
        "ma-task-name": "Controller configuration",
        "ma-task-registry-entry":
          "urn:ietf:lmap:control:http_controller_configuration",
        "ma-task-options": [{"name": "channel",
          "value": "Controller channel"}]
      },
      {
        "ma-task-name": "Controller status and capabilities",
        "ma-task-registry-entry":
          "urn:ietf:lmap:control:http_control_status_and_capabilities",
        "ma-task-options": [{"name": "channel",
          "value": "Controller channel"}]
      },
      {
        "ma-task-name": "Controller instruction",
        "ma-task-registry-entry":
          "urn:ietf:lmap:control:http_controller_instruction",
        "ma-task-options": [{"name": "channel",
          "value": "Controller channel"}]
      }
    ],
    "ma-control-channels": [
      {
        "ma-channel-name": "Controller channel",
        "ma-channel-target": "http://www.example.com/lmap/controller",
        "ma-channel-credientials": { }
      }
    ],
    "ma-control-schedules": [
      {
        "ma-schedule-name": "Controller schedule",
        "ma-schedule-tasks": [
          {
            "ma-schedule-task-name": "Controller configuration",
          },
          {
            "ma-schedule-task-name":
              "Controller status and capabilities",
```

```
           },
           {
             "ma-schedule-task-name": "Controller instruction",
           }
         ],
         "ma-schedule-timing": {
           "ma-timing-name": "hourly randomly",
           "ma-timing-calendar": {
             "ma-calendar-minutes": ["00"],
             "ma-calendar-seconds": ["00"]
           },
           "ma-timing-random-spread": "3600000"
         }
       }
     ],
     "ma-credentials": { }
   }
 }
```

   The above configuration now contacts the Controller randomnly within
   each hour. The following is an example of the Status and
   Capabilities information that is transferred from the MA to the
   Controller.

```
 // Status and Capabilities

 {
   "ma-status-and-capabilities": {
     "ma-agent-id": "550e8400-e29b-41d4-a716-446655440000",
     "ma-device-id": "urn:dev:mac:0024befffe804ff1",
     "ma-hardware": "mfr-home-gateway-v10",
     "ma-firmware": "25637748-rev2a",
     "ma-version": "ispa-v1.01",
     "ma-interfaces": [
       {
         "ma-interface-name": "broadband",
         "ma-interface-type": "PPPoE"
       }
     ],
     "ma-last-task": "",
     "ma-last-report": "",
     "ma-last-instruction": "",
     "ma-last-configuration": "2014-06-08T22:47:31+00:00",
     "ma-supported-tasks": [
       {
```

```
          "ma-task-name": "Controller configuration",
          "ma-task-registry":
            "urn:ietf:lmap:control:http_controller_configuration"
        },,
        {
          "ma-task-name": "Controller status and capabilities",
          "ma-task-registry":
            "urn:ietf:lmap:control:http_control_status_and_capabilities"
        },
        {
          "ma-task-name": "Controller instruction",
          "ma-task-registry":
            "urn:ietf:lmap:control:http_controller_instruction"
        },
        {
          "ma-task-name": "Report",
          "ma-task-registry": "urn:ietf:lmap:report:http_report"
        },
        {
          "ma-task-name": "UDP Latency",
          "ma-task-registry":
            "urn:ietf:ippm:measurement:UDPLatency-Poisson-XthPercMean"
        }
      ]
    }
  }
```

   After fetching the status and capabilties the Controller issues and
   Instruction to the MA to perform a single UDP latency measurement
   task 4 times a day and to report the results immediately.

```
// Instruction

{
  "ma-instruction": {
    "ma-instruction-tasks": [
      {
        "ma-task-name": "UDP Latency",
        "ma-task-registry-entry":
          "urn:ietf:ippm:measurement:UDPLatency-Poisson-XthPercMean",
        "ma-task-options": [
          {"name": "X", "value": "99"},
          {"name":"rate", "value": "5"},
          {"name":"duration", "value": "30.000"},
          {"name":"interface", "value": "broadband"},
```

```
          {"name":"destination-ip",
              "value": {"version":"ipv4", "ip-address":"192.168.2.54"}},
          {"name":"destination-port", "value": "50000"},
          {"name":"source-port", "value": "50000"}
        ],
        "ma-task-suppress-by-default": "TRUE"
      },
      {
        "ma-task-name": "Report",
        "ma-task-registry-entry": "urn:ietf:lmap:report:http_report",
        "ma-task-options": [
          {"name": "report-with-no-data", "value": "FALSE"},
          {"name": "channel", "value": "Collector A"]}
        ],
        "ma-task-suppress-by-default": "FALSE"
      }
    ],
    "ma-report-channels": [
      {
        "ma-channel-name": "Collector A",
        "ma-channel-target": "http://www.example2.com/lmap/collector",
        "ma-channel-credientials": { }
      }
    ],
    "ma-instruction-schedules": [
      {
        "ma-schedule-name": "4 times daily test UDP latency and report",
        "ma-schedule-tasks": [
          {
            "ma-schedule-task-name": "UDP Latency",
            "ma-schedule-destination-tasks": [
              {
                "ma-schedule-task-output-selection": [1],
                "ma-schedule-task-destination-schedule-name":
                  "4 times daily test UDP latency and report",
                "ma-schedule-task-destination-task-configuration-names":
                  "Report"
              }
            ]
          },
          {
            "ma-schedule-task-name": "Report",
          }
        ],

        "ma-schedule-timing": {
          "ma-timing-name": "once every 6 hours",
          "ma-timing-calendar": {
```

```
            "ma-calendar-hours": ["00", "06", "12", "18"],
            "ma-calendar-minutes": ["00"],
            "ma-calendar-seconds": ["00"]
          },
          "ma-timing-random-spread": "21600000"
        }
      }
    ]
  }
}
```

The report task in the Instruction is executed immediately after the UDP test and transfers the following data to the Collector.

```
// Report

{
  "ma-report": {
    "ma-report-date": "2014-06-09T02:30:45+00:00",
    "ma-report-agent-id": "550e8400-e29b-41d4-a716-446655440000",
    "ma-report-tasks": [
      {
        "ma-report-task-name": "UDP Latency",
        "ma-report-task-registry-entry":
          "urn:ietf:ippm:measurement:UDPLatency-Poisson-XthPercMean",
        "ma-report-scheduled-task-options": [
          {"name": "X", "value": "99"},
          {"name":"rate", "value": "5"},
          {"name":"duration", "value": "30.000"},
          {"name":"interface", "value": "broadband"},
          {"name":"destination-ip",
              "value": {"version":"ipv4", "ip-address":"192.168.2.54"}},
          {"name":"destination-port", "value": "50000"},
          {"name":"source-port", "value": "50000"}
        ],
        "ma-report-task-column-labels":
          ["start-time", "conflicting-tasks", "cross-traffic",
              "mean", "min", "max"],
        "ma-report-task-rows":
          ["2014-06-09T02:30:10+00:00", "", "0",
              "20.13", "18.3", "24.1"]
      }
    ]
  }
}
```

The Controller decides that there is a problem with the UDP L:atency test and issues a Suppression Instruction. Since the task is marked as suppressible by default, simply turning on suppression will stop the task being executed in future.

```
    // Suppression

    {
        "ma-instruction": {
          "ma-suppression": {
              "ma-suppression-enabled": "TRUE"
          }
        }
      }
```

Authors' Addresses

    Trevor Burbridge
    BT
    Adastral Park, Martlesham Heath
    Ipswich IP5 3RE
    United Kingdom

    Email: trevor.burbridge@bt.com

    Philip Eardley
    BT
    Adastral Park, Martlesham Heath
    Ipswich IP5 3RE
    United Kingdom

    Email: philip.eardley@bt.com

    Marcelo Bagnulo
    Universidad Carlos III de Madrid
    Av. Universidad 30
    Leganes, Madrid 28911
    Spain

    Email: marcelo@it.uc3m.es

Juergen Schoenwaelder
Jacobs University Bremen
Campus Ring 1
Bremen 28759
Germany

Email: j.schoenwaelder@jacobs-university.de