Network Working Group T. Burbridge
Internet-Draft P. Eardley
Intended status: Standards Track BT
Expires: October 12, 2015 M. Bagnulo
Universidad Carlos III de Madrid
J. Schoenwaelder
Jacobs University Bremen
April 10, 2015

Information Model for Large-Scale Measurement Platforms (LMAP)
draft-ietf-lmap-information-model-05

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted 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."

This Internet-Draft will expire on October 12, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

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 specifies 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 behavior 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 URI. If the MA ID is not provided at this stage then it must be provided by the Controller during Configuration.

3.1.1. Definition of ma-preconfig-obj

```
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..*>;
    [uri                  ma-device-id;]
     credentials          ma-credentials;
} ma-preconfig-obj;
```

The ma-preconfig-obj is essentially a subset of the ma-config-obj described below. The ma-preconfig-obj consists of the following elements:

ma-agent-id: An optional uuid uniquely identifying the measurement agent.

ma-control-tasks: A collection of tasks objects.

ma-control-channels: A collection of channel objects.

ma-control-schedules: A collection of scheduling objects.

ma-device-id: An optional identifier for the device.

ma-credentials: The security credentials used by the measurement agent.

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 schedules 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.

3.2.1. Definition of ma-config-obj

```
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..*>;
   [uri                  ma-device-id;]
    credentials          ma-credentials;
   [string               ma-group-id;]
   [boolean              ma-report-agent-id;]
   [int                  ma-controller-lost-timeout;]
} ma-config-obj;
```

The ma-config-obj consists of the following elements:

ma-agent-id: A uuid uniquely identifying the measurement agent.

ma-control-tasks: A collection of task objects.

ma-control-channels: A collection of channel objects.

ma-control-schedules: A collection of scheduling objects.

ma-device-id: An optional identifier for the device.

ma-credentials: The security credentials used by the measurement agent.

ma-group-id: An optional identifier of the group of measurement agents this measurement agent belongs to.

ma-report-agent-id: An optional flag indicating whether the identifier (ma-agent-id) should be included in reports.

ma-controller-lost-timeout: A timer is started after each successful contact with a controller. When the timer reaches the controller-lost-timeout, all schedules will be disabled.

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.

3.3.1. Definition of ma-instruction-obj

```
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;
```

An ma-instruction-obj consists of the following elements:

ma-task-obj: A possibly empty collection of task objects.

ma-channel-obj: A possibly empty collection of channel objects.

ma-schedule-obj: A possibly empty collection of schedule objects.

ma-suppression-obj: A suppression object.

3.3.2. Definition of ma-suppression-obj

```
object {
    boolean            ma-suppression-enabled;
   [boolean            ma-suppression-stop-running;]
   [datetime           ma-suppression-start;]
   [datetime           ma-suppression-end;]
   [string             ma-suppression-task-names<0..*>;]
   [string             ma-suppression-schedule-names<0..*>;]
} ma-suppression-obj;
```

The ma-suppression-obj consists of the following elements:

ma-suppression-enabled: A boolean indicating whether suppression is enabled or not. The default value is false.

ma-suppression-stop-running: An optional boolean indicating whether suppression will stop any running tasks. The default value for this boolean is false.

ma-suppression-start: The optional date and time when suppression starts. The default value is 'immediate'.

ma-suppression-end: The optional date and time when suppression ends. The default value is 'indefinite'.

ma-suppression-task-names: An optional and possibly empty collection of task names. If not present, this defaults to all tasks.

ma-suppression-schedule-names: An optional and possibly empty collection of schedule names. If not present, this defaults to all schedules.

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.

3.4.1. Definition of ma-log-obj

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

The ma-log-obj models the generic aspects of a logging object and consists of the following elements:

ma-log-agent-id: A uuid uniquely identifying the measurement agent.

ma-log-event-time: The date and time of the event reported in the logging object.

ma-log-code: A machine readable code describing the event.

ma-log-description: A human readable description of the event.

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.

3.5.1. Definition of ma-status-obj

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

The ma-status-obj provides status information about the measurement agent and consists of the following elements:

ma-agent-id: A uuid uniquely identifying the measurement agent.

ma-device-id: A URI identifying the device.

ma-hardware: A description of the hardware of the device the measurement agent is running on.

ma-firmware: A description of the firmware of the device the measurement agent is running on.

ma-version: The version of the measurement agent.

ma-interfaces: A list of network interfaces available on the device.

ma-last-started: The date and time the measurement agent last started.

ma-task-status: An optional list of status objects for each supported task.

3.5.2. Definition of ma-task-status-obj

```
object {
    string              ma-task-name;
    [uri                ma-task-registry-entry;]
    [string             ma-task-role<0..*>;]
    datetime            ma-task-last-invocation;
    datetime            ma-task-last-completion;
    int                 ma-task-last-status;
    string              ma-task-last-message;
    datetime            ma-task-last-failed-completion;
    int                 ma-task-last-failed-status;
    string              ma-task-last-failed-message;
} ma-task-status-obj;
```

The ma-task-status-obj provides status information about a task and consists of the following elements:

ma-task-name: A name uniquely identifying a task.

ma-task-registry-entry: An optional URI identifying the nature of the task.

ma-task-role: An optional and possibly empty list of roles of a task.

```
ma-task-last-completion:        The date and time of the last
                                completion of this task.

ma-task-last-status:            The status code returned by the last
                                execution of this task.

ma-task-last-message:           The status message produced by the
                                last execution of this task.

ma-task-last-failed-completion: The date and time of the last failed
                                completion of this task.

ma-task-last-failed-status:     The status code returned by the last
                                failed execution of this task.

ma-task-last-failed-message:    The status message produced by the
                                last failed execution of this task.
```

3.5.3. Definition of ma-interface-obj

```
object {
    string              ma-interface-name;
    string              ma-interface-type;
   [int                 ma-interface-speed;]
   [string              ma-interface-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;
```

The ma-interface-obj provides status information about network interfaces and consists of the following elements:

```
ma-interface-name:                A name uniquely identifying a
                                  network interface.

ma-interface-type:                The type of the network interface.

ma-interface-speed:               An optional indication of the speed
                                  of the interface (measured in bits-
                                  per-second).

ma-interface-link-layer-address:  An optional link-layer address of
                                  the interface.

ma-interface-ip-addresses:        An optional list of IP addresses
                                  assigned to the interface.
```

ma-interface-gateways: An optional list of gateways assigned to the interface.

ma-interface-dns-servers: An optional list of DNS servers assigned to the interface.

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 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.

3.6.1. Definition of ma-report-obj

```
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;
```

The ma-report-obj provides the meta-data of a single report and consists of the following elements:

ma-report-date: The date and time when the report was sent to a collector.

ma-report-agent-id: An optional uuid uniquely identifying the measurement agent.

ma-report-group-id: An optional identifier of the group of measurement agents this measurement agent belongs to.

ma-report-tasks: An optional and possibly empty list of tasks result objects.

3.6.2. Definition of ma-report-task-obj

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

The ma-report-task-obj provides the meta-data of a result report of a single task. It consists of the following elements:

ma-report-task-name: A name uniquely identifying the task that produced the results being reported.

ma-report-task-registry-entry: An optional URI identifying the type of task.

ma-report-task-scheduled-task-options: An optional list of task options provided by the scheduling object.

ma-report-task-cycle-id: An optional measurement cycle identifier.

ma-report-task-column-labels: A possibly empty list of column labels.

ma-report-task-rows: A possibly empty list of result rows.

3.6.3. Definition of ma-report-row-obj

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

The ma-report-row-obj represents a result row and consists of the following elements:

ma-report-result-start-time: The date and time of the start of the measurement task that produced the reported result values.

ma-report-result-end-time: An optional date and time indicating when the measurement task that produced the reported result values finished.

ma-report-result-conflicts: A possibly empty set of names of task that might have impacted the measurement being reported.

ma-report-result-values: A possibly empty set of result values.

3.7. Common Objects: 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.

3.7.1. Definition of ma-schedule-obj

```
object {
    string              ma-schedule-name;
    ma-action-obj       ma-schedule-actions<0..*>;
    ma-timing-obj       ma-schedule-timing;
} ma-schedule-obj;
```

The ma-schedule-obj is the main scheduling object. It consists of the following elements:

ma-schedule-name: A name uniquely identifying a scheduling object.

ma-schedule-actions: A possibly empty list of actions to invoke when the schedule fires.

ma-schedule-timing: A timing object indicating when the schedule fires.

3.7.2. Definition of ma-action-obj

```
object {
    string                    ma-action-name;
    string                    ma-action-task-name;
   [ma-option-obj             ma-action-task-options<0..*>];
   [ma-action-dest-obj    ma-action-destinations<0..*>;]
 } ma-action-obj;
```

The ma-sched-action-obj models an a task together with its schedule specific options and destination tasks. It consists of the following elements:

ma-action-name: A name uniquely identifying an action of a scheduling object.

ma-action-task-name: A name identifying the task to be invoked by the action.

ma-action-task-options: An optional and possibly empty list of options (name-value pairs) that are passed to the task by appending them to the options configured for the task object.

ma-action-destinations: An optional and possibly empty list of destination actions that consume output produced by this action.

3.7.3. Definition of ma-action-dest-obj

```
object {
    string    ma-action-dest-schedule-name;
    string    ma-action-dest-action-name;
   [int       ma-action-dest-action-outputs<0..*>;]
} ma-action-dest-obj;
```

The ma-action-dest-obj defines to which subsequent actions output produced by an action should be sent to. It consists of the following elements:

ma-action-dest-schedule-name: A name identifying a schedule object.

ma-action-dest-action-name: A name identifying an action within a schedule object.

ma-action-dest-action-outputs: An optional and possibly empty list of task outputs. If not present, the element defaults to all outputs.

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 and 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-action-dest-obj with the output selection as [1,2] and the destination action 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 ----/
```

3.8. Common Objects: 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..

3.8.1. Definition of ma-channel-obj

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

The ma-channel-obj consists of the following elements:

ma-channel-name: A unique name identifying the channel object.

ma-channel-target: A URL identifying the target channel endpoint.

ma-channel-credentials: The security credentials needed to establish a secure channel.

ma-channel-interface-name: An optional name of the network interface to be used. If not present, the system will select a suitable interface.

3.9. Common Objects: 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.

3.9.1. Definition of ma-task-obj

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

The ma-task-obj defines a task that can be invoked. A task can be referenced via its name and it contains an URI to link to a registry or a local specification of the task. Options allow the configuration of task parameter (in the form of name-value pairs). The ma-task-obj consists of the following elements:

ma-task-name: A name uniquely identifying a task object.

ma-task-registry-entry: A URI identifying the type of task.

ma-task-options: A optional and possibly empty list of options (name-value pairs) that are passed to the task.

ma-task-suppress-by-default: A boolean flag indicating whether the task will be suppressed by default. The default value of the flag is true.

ma-task-cycle-id: An optional measurement cycle identifier that can be used to identify set of measurement results that have been produced by tasks with comparable options.

3.9.2. Definition of ma-option-obj

```
object {
    string       ma-option-name;
   [object       ma-option-value;]
} ma-option-obj;
```

The ma-option-obj models a name-value pair and consists of the following elements:

ma-option-name: The name of the option.

ma-option-value: The optional value of the option.

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.

- o Channel is used to specify the details of an endpoint for Control or Reporting Task communications and is detailed elsewhere in this document. The common option name for specifying the channel is "channel".

- o Role is used to specify which Role the task should be performing (as defined in the registry) if multiple roles are available. The common option name for specifying the role is "role".

3.10. Common Objects: 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].

3.10.1. Definition of ma-timing-obj

```
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;]
} ma-timing-obj;
```

The ma-timing-obj is the main timing object. Timing objects are identified by a name. The timing object itself contains a more specific timing object. These objects are further described below.

The ma-timing-obj also includes an optional uniform random spread in milliseconds that can be used to randomize the start times of scheduled tasks. The ma-timing-obj consists of the following elements:

ma-timing-name: The name uniquely identifies a timing object. Schedules refer to timing objects by this name.

ma-timing-periodic: The ma-timing-periodic is present for periodic timing objects.

ma-timing-calendar: The ma-timing-calendar is present for calendar timing objects.

ma-timing-one-off: The ma-timing-one-off is present for one-off timing objects.

ma-timing-immediate: The ma-timing-immediate is present for immediate timing objects.

ma-timing-startup: The ma-timing-startup is present for startup timing objects.

ma-timing-random-spread: The optional ma-timing-random-spread adds a random delay defined in milliseconds to the timing object.

3.10.2. Definition of ma-periodic-obj

```
object {
   [datetime        ma-periodic-start;]
   [datetime        ma-periodic-end;]
    int             ma-periodic-interval;
} ma-periodic-obj;
```

The ma-periodic-obj timing object has an optional start and an optional end time plus a periodic interval. Tasks scheduled using an ma-periodic-obj are started periodically between the start and end time. The ma-periodic-obj consists of the following elements:

ma-periodic-start: The optional date and time at which tasks scheduled using this object are first started. If not present it defaults to immediate.

ma-periodic-end: The optional date and time at which tasks scheduled using this object last started. If not present it defaults to indefinite.

ma-periodic-interval: The interval defines the time in milliseconds between two consecutive starts of tasks.

3.10.3. Definition of ma-calendar-obj

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).

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.

```
object {
   [datetime        ma-calendar-start;]
   [datetime        ma-calendar-end;]
   [string          ma-calendar-months<0..*>;]
   [string          ma-calendar-days-of-week<0..*>;]
   [string          ma-calendar-days-of-month<0..*>;]
   [string          ma-calendar-hours<0..*>;]
   [string          ma-calendar-minutes<0..*>;]
   [string          ma-calendar-seconds<0..*>;]
   [int             ma-calendar-timezone-offset;]
} ma-calendar-obj;
```

ma-calendar-start: The optional date and time at which tasks scheduled using this object are first started. If not present it defaults to immediate.

ma-calendar-end: The optional date and time at which tasks scheduled using this object last started. If not present it defaults to indefinite.

ma-calendar-months: The optional set of months (1-12) on which tasks scheduled using this object are started. The wildcard * means all months. If not present, it defaults to no months.

ma-calendar-days-of-week: The optional set of days of a week ("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun") on which tasks scheduled using this object are started. The wildcard * means all days of teh week. If not present, it defaults to no months.

ma-calendar-days-of-month: The optional set of days of a months (1-31) on which tasks scheduled using this object are started. The wildcard * means all days of a months. If not present, it defaults to no days.

ma-calendar-hours: The optional set of hours (0-23) on which tasks scheduled using this object are started. The wildcard * means all hours of a day. If not present, it defaults to no hours.

ma-calendar-minutes: The optional set of minutes (0-59) on which tasks scheduled using this object are started. The wildcard * means all minutes of an hour. If not present, it defaults to no hours.

ma-calendar-seconds: The optional set of seconds (0-59) on which tasks scheduled using this object are started. The wildcard * means all seconds of an hour. If not present, it defaults to no seconds.

ma-calendar-timezone-offset: The optional timezone offest in hours. If not present, it defaults to the system's local timezone..

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

3.10.4. Definition of ma-one-off-obj

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

The ma-one-off-obj timing object specifies a fixed point in time. Tasks scheduled using an ma-one-off-obj are started once at the specified date and time. The ma-one-off-obj consists of the following elements:

ma-one-off-time: The date and time at which tasks scheduled using this object are started.

3.10.5. Definition of ma-immediate-obj

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

The ma-immediate-obj timing object has no further information elements. Tasks scheduled using an ma-immediate-obj are started as soon as possible.

3.10.6. Definition of ma-startup-obj

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

The ma-startup-obj timing object has no further information elements. Tasks scheduled using an ma-startup-obj are started at MA initiation time.

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 of Broadband Performance (LMAP)", draft-ietf-lmap-framework-12 (work in progress), March 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-02 (work in progress), February 2015.

[I-D.ietf-lmap-yang]
Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for LMAP Measurement Agents", draft-ietf-lmap-yang-00 (work in progress), April 2015.

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

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