                    Large MeAsurement Platform Protocol
                        draft-bagnulo-lmap-http-03

Abstract

   This documents specifies the LMAP protocol based on HTTP for the
   Control and Report in Large Scale Measurement Platforms.

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 March 14, 2015.

Table of Contents

1. Introduction

   A Large MeAsurement Platform (LMAP) is an infrastructure deployed in
   the Internet that enables performing measurements from a very large
   number of vantage points.

   The main components of a LMAP are the following:

   o The Measurement Agents (MAs): these are the processes that perform
      the measurements. The measurements can be both active or passive
      measurements.

o The Controller: this is the element that controls the MAs. In
   particular it provides configuration information and it instructs
   the MA to perform a set of measurements.
o The Collector: this is the repository where the MAs send the
   results of the measurements that they have performed.

These and other terms used in this document are defined in
[I-D.ietf-lmap-framework]. We only include the definition of the
main elements in this document so it is self-contained and can be
read without the need to consult other documents. The reader is
referred to the terminology draft for further details.

In order for a LMAP to work, the following protocols are required:

o Measurement protocols: These are the protocols used between the MA
   and the Measurement Peer in active measurements. These are the
   actual packets being used for the measurement operations.
o Control Protocol. This is the protocol between the Controller and
   the MAs. This protocol is used to convey measurement
   Instruction(s) from the Controller to the MA as well as logging,
   failure and capabilities information from the MA to the
   Controller.
o Report Protocol. This is the protocol between the MAs and the
   Collector. This protocol conveys information about the results of
   the measurements performed by the MA to the Collector.

Both the Control protocol and the Report protocol have essentially
two parts:a transport and a data model. The data model represents
the information about measurement instructions and logging/failure/
capabilities (in the Control protocol) and the information about
measurement results (in the Report protocol) that is being exchanged
between the parties. The transport is the underlying protocol used
to exchange that information. This document specifies the use of
HTTP 1.1 [RFC7230] [RFC7231] [RFC7232] [RFC7233] [RFC7234] [RFC7235]
as a transport for the Control and the Report protocol. This
document also defines the data model for the Control and Report
protocols. The data model described in this document follows the
information model described in [I-D.ietf-lmap-information-model].
The Measurement protocols are out of the scope for this document.

At this stage, the goal of this document is to explore different
options that can be envisioned to use the HTTP protocol to exchange
LMAP information and to foster discussion about which one to use (if
any). Because of that, the document contains several discussion
paragraphs that explore different alternative approaches to perform
the same function.

2. Overview

   This section provides an overview of the architecture envisioned for
   a LMAP using HTTP as transport protocol. As we described in the
   previous section, a LMAP is formed by a large number of MAs, one or
   more Controllers and one or more Collectors. We assume that before
   the MAs are deployed, it is possible to pre-configure some
   information in them. Typically this includes information about the
   MA itself (like its identifier), security information (like some
   certificates) and information about the Controller(s) available in
   the measurement platform. Once that the MA is deployed it will
   retrieve additional configuration information from the pre configured
   Controller. After obtaining the configuration information, the MA is
   ready to receive Instructions from the Controller and initiate
   measurement tasks. The MA will perform the following operations:

   o It will obtain Instructions from one of the configured
      Controllers. These Instructions include information about the set
      of measurement tasks to be performed, a schedule for the execution
      of the measurements as well as a set of report channels. This
      information is downloaded by the MA from the Controller. The MA
      will periodically check whether there are new Instructions
      available from the Controller. This document specifies how the MA
      uses the HTTP protocol to retrieve information from the
      Controller.
   o The MA will execute measurement tasks either by passively
      listening to traffic or by actively sending and receiving
      measurement packets. How this is done is out of the scope of this
      document.
   o After one or more measurements have been performed, the MA reports
      the results to the Collector. The timing of these uploads is
      specified in the measurement Instruction i.e. each measurement
      specified in a measurement Instruction contains a report
      information, defining when the MA should report the results back
      to the Collector. This document specifies how the MA uses the
      HTTP protocol to upload the measurement results to the Collector.
   o In addition, the MA will periodically report back to the
      Controller information about its capabilities (like the number of
      interfaces it has, the corresponding IP addresses, the set of
      measurement methods it supports, etc) and also logging information
      (whether some of the requested measurement tasks failed and
      related information).

3. Naming Considerations

   In this section we define how the different elements of the LMAP
   architecture are identified and named.

The Controller and the Collectors can be assumed to have both an IP
address and a Fully Qualified Domain Name (FQDN). It is natural to
use these as identifiers for these elements. In this document we
will use FQDNs, but IP addresses can be used as well.

The MAs on the other hand, are likely to be executed in devices
located in the end user premises and are likely to be located behind
a NAT box. It is reasonable to assume they have neither a public IP
address nor a FQDN. We propose then that the MAs are identified
using an Universally Unique IDentifier URN as defined in RFC 4122
[RFC4122]. In particular each MA has a version 4 UUID, which is
randomly or pseudo randomly generated.

DISCUSSION:

   MA ID Configuration: Some open issues related to this are: a)
   whether the MA ID is configured before of after the MA is
   deployed, b) if configured after deployment whether the MA ID is
   generated locally and posted or fetched from the Controller and c)
   whether this is within the scope of this (or other) specification
   if any. These issues seem also to be related to the nature of the
   MA platform (wether the MA is a software downloaded into a general
   purpose device or it is a special purpose hardware box). Consider
   the case that the MA is located in a special purpose hardware box,
   then having the MA ID pre configure before deployment requires a
   per device customization that is expensive. It would be more
   costly efficient to reuse an existent (hopefully) unique
   identifier available in the hardware (such as a MAC address) to
   serve as a one-time pre configured identifier to be used to fetch
   (or post a self generated) the MA ID from the Controller once the
   MA is deployed. The requirement for such one-time identifier is
   that they must be unique (which is not always true for the MACs).
   About the local generation of the MA ID (as opposed to fetch it
   from the Controller), the generation process performed in the MA
   MUST be idempotent, i.e. if the MA was factory-reset then the
   server would still see it with the same MA ID when it came back
   up. This is probably easier to achieve if it is generated in the
   Controller and then fetched by the MA. Finally, it is not clear
   at this stage if this needs to be specified in this document or in
   the information model document or left open to the implementers.
   Group identifiers. In some cases, like the case of measurements
   in mobile devices, it may be important because of privacy
   considerations for the MA not to have a unique identifier. It is
   possible then to assign "Group identifiers" to a set of devices
   that share relevant characteristics from the measurement
   perspective (e.g. devices from the same operator, with the same
   type of contract or other relevant feature). In this case, the
   MAs within the same group would retrieve common measurement

Instructions from the controller by presenting the same Group ID
and would report results including the Group ID in the report.
This would imply that it would not be possible for the platform to
correlate specific measurement data with any given MA. The
downside of this is that some MAs may be over-represented while
other under-represented in the measurement data and it would not
be possible to detect this case (for instance a given MA may have
reported 20 results while another one only one). In order to deal
with this issue, the MA behaviour must be programmed accordingly
(e.g. the MA should not perform more than one measurement every
given period of time).In addition, it should be noted that privacy
is only achieved in a holistic way. This means that really
anonymity of the MA is incompatible with strong authentication.
In particular, if a measurement platform's goal is to keep MAs
anonymous, it cannot require any form of strong authentication
(other than weak group authentication e.g. a password shared by a
group), which has security implications. In particular, the
threat for report forgery (i.e. enabling an attacker to submit
forged reports as discussed in the security considerations)
increases.

   There are additional naming considerations related to:

   o The measurements. In order to enable a Controller to properly
      convey a measurement schedule, it must be possible for the
      Controller to specify a measurement to be performed while
      providing the needed input parameters. While this is critical, it
      is out of the scope of this document. There is a proposed
      registry for metrics/measurements in
      [I-D.bagnulo-ippm-new-registry-independent])
   o The resources being exchanged, namely, the configuration
      information, the measurement Instructions and the reports. These
      are being discussed in the upcoming sections.

4. Information model

   The information model for LMAP is described
   [I-D.ietf-lmap-information-model]. It contains basically two models
   one for the control information (i.e. the Instructions from the
   Controller to the MA) and a model for the Report information. We
   briefly describe their overall structure here.

   The control information (or Instruction) has the following five
   elements:

   o The Set of Measurement Task Configurations: This element defines
      the measurements/test that the MA will perform without defining
      the schedule when they will be performed.

o The Set of Report Channels: This element defines the set of
   collectors as well as the reporting schedules for the reports.
o The Set of Measurement Schedules for Repeated Tasks: defines the
   schedules for the repeated measurements, by referencing the
   measurement tasks defined in the second element.
o Suppression information

Summary of Report information model here.

Summary of Capability and Status information model here.

Summary of Logging information model here.

5. Transport protocol

5.1. Pre-configured information

   As we mentioned earlier, the MAs contain pre-configured information
   before being deployed. The pre-configured information is the
   following:

   o The UUID for the MA. This should be pre-configured so that the
      Controller is aware of the MA and can feed configuration
      information and measurement Instructions to it.
   o Information about one or more Controllers. The MA MUST have
      enough information to create the URL for the Instruction
      resources. This includes the the FQDN of each of the Controller
      or the IP addresses of the Controller, as well as the well-known
      path prefix and its identifier.
   o The certificate for the Certification authority that is used in
      the platform to generate the certificates for the Controller and
      the Collector. See the Security considerations section below.
   o The security related information for the MA (it can be a
      certificate for the MA and the corresponding private key, or
      simply a key/password depending on the security method used, see
      the security considerations section below).

5.2. Control Protocol

   The Control protocol is used by the MA to retrieve Instruction
   information from the Controller. In this section we describe how to
   use HTTP to transport Instructions. The Instruction information is
   structured as defined in the LMAP Information model
   [I-D.ietf-lmap-information-model] as described in the previous
   section. The MA uses the Control protocol to retrieve all the
   resources described above, namely, the Agent information, the Set of
   Measurement Task Configurations, the Set of Report Channels, the Set
   of Measurement Schedules for Repeated Tasks and the Set of

Measurement Schedules for Isolated Tasks. The main difference from
the HTTP perspective is that the MA MUST have the URL for the Agent
Information resource pre-configured as described in the previous
section, while the URLs for all the other resources are contained in
the Agent Information resource itself.

5.2.1. Retrieving Instructions

In order to retrieve the Instruction resources from the Controller
the MA can use either the GET or the POST method using the
corresponding URL.

5.2.1.1. Using the GET method

One way of using the GET method to retrieve configuration information
is to explicitly name the configuration information resources and
then apply the GET method. The MA retrieves its Instruction when it
is first connected to the network and periodically after that. The
frequency for the periodical retrieval is contained in the Agent
Information (???).

The URL for the Agent Information resource is formed as the FQDN of
the Controller, a well-known path prefix and the MA UUID. The well-
known path prefix is /.well-known/lmap/ma-info. The URL for the
remaining resources that compose the Instruction are contained in the
Agent Information.

Agent Information retrieval: In order to retrieve the Agent
information the MA uses the HTTP GET method follows:

    GET /.well-known/lmap/ma-info/ < ma-iid> HTTP/1.1
    Host: FQDN or IP of the Controller
    Accept: application/json (as per [RFC7159])

The Agent Information should contain the Configuration Retrieval
Schedule (i.e. how often the MA should retrieve configuration
information) and also the Measurement Instruction Retrieval Schedule
(i.e. how often the MA should retrieve the Measurement Instruction
from the Controller). COMMENT: this is missing from the Data Model

The retrieval of the remaining resources of the Instruction using the
GET method is analogous, only that the URL is extracted from the
Agent Information file rather than constructed with pre-configured
information.

The format for the response should be described here

Periodical Instruction retrieval: After having downloaded the initial
Instruction information, the MA will periodically look for updated
Instruction information. The frequency with which the MA polls for
the new Instructions from the Controller is contained in the last
Agent Information downloaded. In order to retrieve the Agent
Information, the MA uses the GET method as follows:

```
GET /.well-known/lmap/ma-info/ma-iid/ HTTP/1.1
Host: FQDN or IP of the Controller
Accept: application/json (as per [RFC7159])
If-None-Match: the eTag of the last retrieved Agent Information
(an alternative option here is to use If-Modified-Since, not sure
which one is best)
```

For the other Instruction resources, the GET method is applied in the
same way just that the URL used are the ones retrieved in the last
Agent Information.

The format for the response should be described here

Alternatively, instead of explicitly naming the Instruction resources
for each MA, it is possible to perform a query using the GET method
as well. In this case, the MA could perform a GET for the following
URI http://controller.example.org/?ma=maid & q=ma-info (similar
queries can be constructed for the other Instruction resources). (I
am not sure how to express in this case the condition that the MA
wishes to retrieve the configuration if it is newer than the last one
it downloaded.)

5.2.1.2. Using the POST method

An alternative to retrieve Instruction resources is to use the POST
method to perform a query (similar to the query using GET). In this
case there is no explicit naming of the Instruction information of
each MA, but a general Instruction resource and the POST method is
used to convey a query for the Instruction information of a
particular MA. For the case of the Agent Information resource, this
would look like as follows:

```
POST /.well-known/lmap/ma-info/ma-iid/ HTTP/1.1
Host: controller.example.com
Content-Type: application/lmap-maid+json
Accept: application/lmap-config+json
{
"ma-id" : "550e8400-e29b-11d4-a716-446655440000",
}
```

The reply for this query would contain the actual configuration
information as follows:

```
HTTP/1.1 200 OK
Content-Length: xxx
Content-Type: application/lmap-config+json
{
// whatever config goes here
}
```

In this case, the URLs contained in the Agent information can be
generic and not MA specific, since the MA will use the POST method
including its own identifier when retrieving the Instruction
resources.

The argument for this approach is that this is much more extensible
since the POST can carry complex information and there is no need to
"press" arguments into the strict hierarchy of URIs.

We need to describe how to use this to retrieve newer information in
the periodic case.

5.2.2. Handling communication failures

The cases that the MA is unable to retrieve the Instructions are
handled as follows:

o The MA will use a timeout for the communication of TIMEOUT
   seconds. The value of TIMEOUT MUST be configurable via the
   aforementioned Configuration Information retrieval protocol. The
   default value for the TIMEOUT is 3 seconds. If after the timeout,
   the communication with the Controller has not been established,
   the MA will retry doing an exponential backoff and doing a round
   robin between the different Controllers it has available.
o If a HTTP error message (5xx) is received from the Controller as a
   response to the GET request, the MA will retry doing an
   exponential back-off and doing a round robin between the different
   Controllers it has available. The 5xx error codes indicate that
   this Controller is currently incapable of performing the requested
   operation.

5.2.3. Pushing Information from the Controller to the MA

The previous sections described how the MA periodically polls the
Controller to retrieve Instruction information. The frequency of the
downloads is configurable. The question is whether this is enough or
a mechanism for pushing Instruction information is needed. Such
method would enable to contact the MA in any moment and take actions

like triggering a measurement right away or for instance to stop an
ongoing measurement (e.g. because it is disturbing the network).
The need for such a mechanism is likely to depend on the use case of
the platform. Probably the ISP use case is more likely to require
this feature than the regulator/benchmarking use case. It is
probably useful then to provide this as an optional feature.

The main challenge in order to provide this feature is that the MAs
are likely to be placed behind NATs, so it is not possible for the
Controller to initiate a communication with the MA unless there is a
binding in the NAT to forward the packets to the MA. There are
several options that can be considered to enable this communication:

o The MA can use one of the NAT control protocols, such as PCP or
   UPNP. If this approach is used, the MA will create a binding in
   the NAT opening a hole. After that, the MA should inform the
   Controller about which is the IP address and port available for
   communication. It would be possible to re-use existing protocols
   to forward this information. The problem with this is that the
   NAT may not support these protocols or they may not be activated.
   In any case, a solution should try to use them in the case they
   are available.
o If it is not possible to use a NAT control protocol, then the MA
   can open a hole in the NAT by establishing a connection to the
   Controller and keeping it open. This allows the Controller to
   push information to the MA through that connection. One concern
   with this approach is that the MA is playing the role of the
   client and the Controller is playing the role of the server (the
   MA is initiating the TCP connection), but it would be the
   Controller who would use the PUT method towards the MA reversing
   the roles. An alternative approach is that the MA has a long
   running GET pending which is answered by the server if the
   measurement Instruction changes (or the server times out, in which
   case the MA restarts the long running GET. More discussion is
   needed about whether one of these options is acceptable or not.
   In addition, this would imply that the Controller should maintain
   as many open sessions as MAs it is managing, which imposes
   additional burden in the Controller. There are security
   considerations as well, but these are covered in the Security
   Considerations section below.

5.3. Report protocol

   The MA after performing the measurements reports the results to a
   collector. There can be more than one collector within a LMAP
   framework. Each collector is identified by its FQDN or IP address
   which is retrieved as part of the Agent information from a pre-
   configured controller as previously discussed. The number of

Collectors that the MA uploads the results to as well as the schedule
when it does so is defined in the measurement Instruction previously
downloaded from the Controller. The MA themselves are identified by
a UUID.

There are two options that can be considered for the MA to upload
reports to the Collector either to use the PUT method or to use the
POST method.

If the PUT method option is used, then the MA need to perform the PUT
method using an explicit name for the report resource it is
transferring to the Collector. The name of the resource is contained
in the Agent Information previously retrieved by the MA

The other option is for the MA to use the POST method to upload the
measurement reports to one or more Collectors. In this case,, the
POST message body can contain the identifier of the MA and additional
information describing the report in addition to the report itself.

One argument to consider is that PUT is idempotent. This means that
if the network is bad at some point and the MA is not sure whether
its request made it through, it can send it a second (or nth) time,
and it is guaranteed that the request will have exactly the same
effect as sending it for the first time. POST does not by itself
guarantee this. This can be achieved by verifying the report data
itself, and contrast it with data already stored int he Collector
database.

5.3.1. Handling communication failures

The MA will use a timeout for the communication with the Collector of
TIMEOUT seconds. The value of TIMEOUT MUST be configurable via the
aforementioned Configuration Information retrieval protocol. The
default value for the TIMEOUT is 3 seconds.

If the MA is uploading the report to several Collectors and it
manages to establish the communication before TIMEOUT seconds with at
least one of them, but not with one or more of the other Collectors,
then the MA gives up after TIMEOUT seconds and it MAY issue an alarm.
The definition of how to do that operation is out of the scope of
this document.

If the MA is uploading the report to only one Collector, and it does
not manages to establish a communication before TIMEOUT seconds, then
it retry doing an exponential backoff and doing a round robin between
the different Collectors it has available.

   Similarly, if an HTTP error message (5xx) is received from the
   Collector as a response to the PUT request, the MA will retry doing
   an exponential backoff and doing a round robin between the different
   Collectors it has available. The 5xx error codes indicate that this
   Collector is currently incapable of performing the requested
   operation.

   In order to support this, the information model must express the
   difference between a report sent to multiple collectors and multiple
   collectors used for fallback.

6. LMAP Data Model

   This section will contain the data model in json.

6.1. Timing Information

   An example immediate timing object with no defined randomness is
   shown below:

   {
     "timings": [
       {
         "id": 1,
         "ma_periodic_end": 1410017611,
         "ma_periodic_interval": 3600000,
         "ma_periodic_start": 1410017613,
         "ma_randomness_spred": 0,
         "ma_timing_name": "hourly"
       },
       {
         "id": 3,
         "ma_periodic_end": 1410017611,
         "ma_periodic_interval": 86400,
         "ma_periodic_start": 1410017613,
         "ma_randomness_spred": 0,
         "ma_timing_name": "daily"
       },
       {
         "id": 2,
         "ma_periodic_end": 1410017611,
         "ma_periodic_interval": 3600000,
         "ma_periodic_start": 1410017613,
         "ma_randomness_spred": 0,
         "ma_timing_name": "hourly"
       },
       {

```
      "id": 4,
      "ma_calendar_days_of_month": "",
      "ma_calendar_days_of_week": "tuesday, thursday, sunday",
      "ma_calendar_end": 1410017613,
      "ma_calendar_hours": "18",
      "ma_calendar_minutes": "04",
      "ma_calendar_months": "",
      "ma_calendar_seconds": "42",
      "ma_calendar_start": 1410017612,
      "ma_calendar_timezone_offset": 2,
      "ma_randomness_spred": 0,
      "ma_timing_name": "tuesday-thursday-sunday"
    },
    {
      "id": 5,
      "ma_calendar_days_of_month": "",
      "ma_calendar_days_of_week": "",
      "ma_calendar_end": 1410017619,
      "ma_calendar_hours": "0, 6 12 18",
      "ma_calendar_minutes": "0",
      "ma_calendar_months": "",
      "ma_calendar_seconds": "0",
      "ma_calendar_start": 1410017612,
      "ma_calendar_timezone_offset": 2,
      "ma_randomness_spred": 21600000,
      "ma_timing_name": "once-every-six-hours"
    },
    {
      "id": 6,
      "ma_one_off_time": 1410017613,
      "ma_randomness_spred": 0,
      "ma_timing_name": "immediate"
    },
    {
      "id": 7,
      "ma_one_off_time": 1410017613,
      "ma_randomness_spred": 0,
      "ma_timing_name": "immediate"
    },
    {
      "id": 8,
      "ma_randomness_spred": 12345,
      "ma_timing_name": "startup"
    }
  ]
}
```

6.2. Channels

   An example channel object using the aforementioned timing object is
   shown below:

```
{
  "channels": [
    {
      "id": 1,
      "ma_channel_credentials": "MIIFEzCCAvsCAQEwDQYJ...",
      "ma_channel_interface_name": "eth0",
      "ma_channel_name": "default-collector-channel",
      "ma_channel_target": "collector.example.org"
    },
    {
      "id": 2,
      "ma_channel_credentials": "MIIFEzCCAvsCAQEwDQYJ...",
      "ma_channel_interface_name": "eth0",
      "ma_channel_name": "default-controller-channe",
      "ma_channel_target": "controller.example.org"
    }
  ]
}
```

6.3. Configuration

   An example config object using the aforementioned channel objects is
   shown below:

```
{
  "config": [
    {
      "id": 1,
      "ma_agent_id": "550e8400-e29b-41d4-a716-446655440000",
      "ma_channel_name": "default-controller-channel",
      "ma_control_channel_fail_tresh": "10",
      "ma_credentials": "MIIFEzCCAvsCAQEwDQYJ...",
      "ma_device_id": "01:23:45:67:89:ab",
      "ma_group_id": "550e8400-e29b-41d4-a716-446655440123",
      "ma_report_ma_id_flag": "1"
    }
  ]
}
```

6.4. Instruction

   The instruction object is essentially a wrapper around suppression,
   schedule, task, channel objects.

6.5. Measurement Supression

   An example supression object used by the aforementioned instruction
   object is shown below:

   {
     "supression": [
       {
         "id": 1,
         "ma_supression_enabled": 0,
         "ma_supression_end": 0,
         "ma_supression_schedule_names": "icmp-latency-immediate",
         "ma_supression_start": 1410037509,
         "ma_supression_stop_ongoing_task": 0,
         "ma_supression_task_names": "iperf-server"
       }
     ]
   }

6.6. Measurement Task Configurations

   An example task object used by the aforementioned instruction object
   is shown below:

   {
     "tasks": [
       {
         "id": 1,
         "ma_role": "",
         "ma_task_cycle_id": "1",
         "ma_task_name": "udp-latency-test",
         "ma_task_options": "",
         "ma_task_registry_entry": "urn:...",
         "ma_task_supress_default": "true"
       },
       {
         "id": 5,
         "ma_role": "",
         "ma_task_cycle_id": "1",
         "ma_task_name": "reporting-daily",
         "ma_task_options": "",
         "ma_task_registry_entry": "urn:...",
         "ma_task_supress_default": "true"

```
      },
      {
        "id": 2,
        "ma_role": "",
        "ma_task_cycle_id": "1",
        "ma_task_name": "icmp-latency-test",
        "ma_task_options": "",
        "ma_task_registry_entry": "urn:...",
        "ma_task_supress_default": "true"
      },
      {
        "id": 3,
        "ma_role": "",
        "ma_task_cycle_id": "1",
        "ma_task_name": "iperf-server",
        "ma_task_options": "{\\\"name\\\":\\\"role\\\",
          \\\"value\\\":\\\"server\\\"}",
        "ma_task_registry_entry": "server",
        "ma_task_supress_default": "false"
      },
      {
        "id": 4,
        "ma_role": "",
        "ma_task_cycle_id": "1",
        "ma_task_name": "lmap-reporting-task",
        "ma_task_options": "",
        "ma_task_registry_entry": "lmap-reportd",
        "ma_task_supress_default": "true"
      }
    ]
  }
```

6.7. Measurement Schedules

   An example schedule object used by the aforementioned instruction
   object is shown below:

```
   {
     "schedules": [
       {
         "id": 1,
         "ma_sched_channel_interface_select": "0",
         "ma_sched_channel_names": "default-collector-channel",
         "ma_sched_task_downstream_config_names": "reporting-daily",
         "ma_sched_task_output_selection": "1",
         "ma_schedule_name": "reporting-immediate",
         "ma_schedule_task_name": "icmp-latency-test",
         "ma_timing_name": "immediate"
       }
     ]
   }
```

6.8. Logging

   An example log object is shown below:

```
   {
     "logging": [
       {
         "id": 1,
         "ma_log_agent_id": "0e49b32b01fa11e4bcaf10ddb1bd23b5",
         "ma_log_code": "200",
         "ma_log_description": "OK",
         "ma_log_event_time": 1404313752
       }
     ]
   }
```

6.9. Capability and Status

   An example status object is shown below:

```
   {
     "status": [
       {
         "id": 1,
         "ma_agent_id": "c54c284a01ee11e48dd310ddb1bd23b5",
         "ma_condition_code": "8081",
         "ma_condition_text": "Cond_Text",
         "ma_device_id": "urn:dev:mac:0024befffe804ff1",
         "ma_firmware": "4560",
         "ma_hardware": "TL-MR3020",
         "ma_interface_dns_server": "8.8.8.8",
         "ma_interface_gateway": "192.168.1.1",
         "ma_interface_ip_address": "192.168.1.10",
         "ma_interface_name": "eth0",
         "ma_interface_speed": "100Mbps",
         "ma_interface_type": "100baseTX",
         "ma_last_config": "140423245",
         "ma_last_instruction": "140431312",
         "ma_last_measurement": "1404315031",
         "ma_last_report": "1404315053",
         "ma_link_layer_addr": "01:23:45:67:89:ab",
         "ma_task_name": "Report",
         "ma_task_registry": "urn:ietf:lmap:report:http_report",
         "ma_task_role": "Role",
         "ma_version": "Busybox"
       }
     ]
   }
```

6.10. Reporting

   An example report object is shown below:

```
   {
     "reporting": [
       {
         "id": 1,
         "ma_report_agent_id": "550e8400-e29b-41d4-a716-446655440000",
         "ma_report_date": 1404315031,
         "ma_report_group_id": "550e8400-e29b-41d4-a716-446655440123",
         "ma_report_result_conflict_task": "0",
         "ma_report_result_cross_traffic": 20,
         "ma_report_result_end_time": 1404315031,
         "ma_report_result_start_time": 1404315031,
         "ma_report_result_values": "result_values",
         "ma_report_task_column_labels": "\"start-time\",
           \"conflicting-tasks\", \"cross-traffic\", \"mean\",
```

```
            \"min\", \"max\"",
          "ma_role": "",
          "ma_task_cycle_id": "1",
          "ma_task_name": "udp-latency-test",
          "ma_task_options": "",
          "ma_task_registry_entry": "urn:...",
          "ma_task_supress_default": "true"
        },
        {
          "id": 2,
          "ma_report_agent_id": "550e8400-e29b-41d4-a716-446655440000",
          "ma_report_date": 1404315031,
          "ma_report_group_id": "550e8400-e29b-41d4-a716-446655440123",
          "ma_report_result_conflict_task": "0",
          "ma_report_result_cross_traffic": 20,
          "ma_report_result_end_time": 1404315031,
          "ma_report_result_start_time": 1404315031,
          "ma_report_result_values": "result_values",
          "ma_report_task_column_labels": "\"start-time\",
          \"conflicting-tasks\", \"cross-traffic\",
          \"mean\", \"min\", \"max\"",
          "ma_role": "",
          "ma_task_cycle_id": "1",
          "ma_task_name": "icmp-latency-test",
          "ma_task_options": "",
          "ma_task_registry_entry": "urn:...",
          "ma_task_supress_default": "true"
        },
        {
          "id": 3,
          "ma_report_agent_id": "550e8400-e29b-41d4-a716-446655440000",
          "ma_report_date": 1404315031,
          "ma_report_group_id": "550e8400-e29b-41d4-a716-446655440123",
          "ma_report_result_conflict_task": "0",
          "ma_report_result_cross_traffic": 20,
          "ma_report_result_end_time": 1404315031,
          "ma_report_result_start_time": 1404315031,
          "ma_report_result_values": "result_values",
          "ma_report_task_column_labels": "\"start-time\",
            \"conflicting-tasks\", \"cross-traffic\",
            \"mean\", \"min\", \"max\"",
          "ma_role": "",
          "ma_task_cycle_id": "1",
          "ma_task_name": "iperf-server",
          "ma_task_options": "{\\\"name\\\":\\\"role\\\",
            \\\"value\\\":\\\"server\\\"}",
          "ma_task_registry_entry": "server",
          "ma_task_supress_default": "false"
```

```
      },
      {
        "id": 4,
        "ma_report_agent_id": "550e8400-e29b-41d4-a716-446655440000",
        "ma_report_date": 1404315031,
        "ma_report_group_id": "550e8400-e29b-41d4-a716-446655440123",
        "ma_report_result_conflict_task": "0",
        "ma_report_result_cross_traffic": 20,
        "ma_report_result_end_time": 1404315031,
        "ma_report_result_start_time": 1404315031,
        "ma_report_result_values": "result_values",
        "ma_report_task_column_labels": "\"start-time\",
        \"conflicting-tasks\", \"cross-traffic\",
        \"mean\", \"min\", \"max\"",
        "ma_role": "",
        "ma_task_cycle_id": "1",
        "ma_task_name": "lmap-reporting-task",
        "ma_task_options": "",
        "ma_task_registry_entry": "lmap-reportd",
        "ma_task_supress_default": "true"
      },
      {
        "id": 5,
        "ma_report_agent_id": "550e8400-e29b-41d4-a716-446655440000",
        "ma_report_date": 1404315031,
        "ma_report_group_id": "550e8400-e29b-41d4-a716-446655440123",
        "ma_report_result_conflict_task": "0",
        "ma_report_result_cross_traffic": 20,
        "ma_report_result_end_time": 1404315031,
        "ma_report_result_start_time": 1404315031,
        "ma_report_result_values": "result_values",
        "ma_report_task_column_labels": "\"start-time\",
          \"conflicting-tasks\", \"cross-traffic\",
          \"mean\", \"min\", \"max\"",
        "ma_role": "",
        "ma_task_cycle_id": "1",
        "ma_task_name": "reporting-daily",
        "ma_task_options": "",
        "ma_task_registry_entry": "urn:...",
        "ma_task_supress_default": "true"
      }
    ]
  }
```

7. Security considerations

   Large Measurement Platforms may result in a security hazard if they
   are not properly secured. This is so because they encompass a large
   number of MAs that can be managed and coordinated easily to generate
   traffic and they can potentially be used for generating DDoS attacks
   or other forms of security threats.

   From the perspective of the protocols described in this documents, we
   can identify the following threats:

   o Hijacking: Probably the worst threat is that an attacker takes
      over the control of one or more MAs. In this case the attacker
      would be able to instruct the MAs to generate traffic or to
      eavesdrop traffic in their location. It is then critical that the
      MA is able to strongly authenticate the Controller. An
      alternative way to achieve this attack is to alter the
      communication between the Controller and the MAs. In order to
      prevent this form of attack, integrity protection of the
      communication between the Controller and the MAs is required.
   o Polluting: Another type of attack is that an attacker is able to
      pollute the Collectors database by providing false results. In
      this case, the attacker would attempt to impersonate one or more
      MAs and upload fake results in the Collector. In order to prevent
      this, the authentication of the MAs with the Collector is needed.
      An alternative way to achieve this is for an attacker to alter the
      communication between the MA and the Collector. In order to
      prevent this form of attack, integrity protection of the
      communication between the MA and the Collector is needed.
   o Disclosure: Another threat is that an attacker may gather
      information about the MAs and their configuration and the
      Measurement schedules. In order to do that, it would connect to
      the Controller and download the information about one or more MAs.
      This can be prevented by using MA authentication with the
      Controller. An alternative mean to achieve this would be for the
      attacker to eavesdrop the communication between the MA and the
      Controller. In order to prevent this, confidentiality in the
      communication between the MA and the Controller is required.
      Similarly, an attacker may wish to obtain measurement result
      information by eavesdropping the communication between the MA and
      the Collector. In order to prevent this, confidentiality in the
      communication between the MA and the Collector is needed.

   In order to address all the identified threats, the HTTPS protocol
   must be used for LMAP (i.e. using HTTP over TLS). HTTPS provides
   confidentiality, integrity protection and authentication, satisfying
   all the aforementioned needs. Ideally, mutual authentication should
   be used. In any case, server side authentication MUST be used. In

order to achieve that, both the Controller and the Collector MUST
have certificates. The certificate of the CA used to issue the
certificates for the Controller and the Collector MUST be pre
configured in the MAs, so they can properly authenticate them. As
mentioned earlier, ideally, mutual authentication should be used.
However, this implies that certificates for the MAs are needed.
Certificate management for a large number of MAs may be expensive and
cumbersome. Moreover, the major threats identified are the ones
related to hijacking of the MAs, which are prevented by
authenticating the Controller. MAs authentication is needed to
prevent Polluting and Disclosure threats, which are less severe. So,
in this case, alternative (cheaper) methods for authenticating MAs
can be considered. The simplest method would be to simply use the MA
UUID as a token to retrieve information. Since the MA UUID is 128
bit long, it is hard to guess. It would be also possible to use a
password and use the HTTP method for authentication. It is not
obvious that managing passwords for a large number of MAs is easier
than managing certificates though.

An additional security consideration is posed by the mechanism to
push information from the Controller to the MAs. If this method is
used, it would be possible its abuse by an attacker to control the
MAs. This threat is prevented by the use of HTTPS. If HTTPS is used
in the established connection between the MA and the Controller, the
only effect that a packet generated by an external attacker to the MA
or the Controller would be to reset the HTTPS connection, requiring
the connection to be re-established.

It is required in this document that both the Controller and that the
Collector are authenticated using digital certificates. The current
specification allows for the MA to have information about the
certificate of the Certification authority used for generating the
Controller and Collector certificates while the actual certificates
are exchanged in band using TLS. Another (more secure) option is to
perform certificate pinning i.e. to configure in the MAs the actual
certificates rather than the certification authority certificate.
Another measure to increase the security would be to limit the
domains that the FQDNs of the Controller and/or the Collector (e.g.
only names in the exmaple.org domain).

Large scale measurements can have privacy implications, especially in
some scenarios like mobile devices performing measurements. In this
memo we have considered using Group IDs to the MA in order to avoid
the possibility for the platform to track each individual MA that is
feeding results.

8. IANA Considerations

    Registration of the well-known URL

9. Acknowledgments

10. References

10.1. Normative References

    [RFC4122]   Leach, P., Mealling, M., and R. Salz, "A Universally
                Unique IDentifier (UUID) URN Namespace", RFC 4122, July
                2005.

    [RFC7159]   Bray, T., "The JavaScript Object Notation (JSON) Data
                Interchange Format", RFC 7159, March 2014.

    [RFC7230]   Fielding, R. and J. Reschke, "Hypertext Transfer Protocol
                (HTTP/1.1): Message Syntax and Routing", RFC 7230, June
                2014.

    [RFC7231]   Fielding, R. and J. Reschke, "Hypertext Transfer Protocol
                (HTTP/1.1): Semantics and Content", RFC 7231, June 2014.

    [RFC7232]   Fielding, R. and J. Reschke, "Hypertext Transfer Protocol
                (HTTP/1.1): Conditional Requests", RFC 7232, June 2014.

    [RFC7233]   Fielding, R., Lafon, Y., and J. Reschke, "Hypertext
                Transfer Protocol (HTTP/1.1): Range Requests", RFC 7233,
                June 2014.

    [RFC7234]   Fielding, R., Nottingham, M., and J. Reschke, "Hypertext
                Transfer Protocol (HTTP/1.1): Caching", RFC 7234, June
                2014.

    [RFC7235]   Fielding, R. and J. Reschke, "Hypertext Transfer Protocol
                (HTTP/1.1): Authentication", RFC 7235, June 2014.

   [I-D.ietf-lmap-information-model]
             Burbridge, T., Eardley, P., Bagnulo, M., and J.
             Schoenwaelder, "Information Model for Large-Scale
             Measurement Platforms (LMAP)", draft-ietf-lmap-
             information-model-02 (work in progress), August 2014.

10.2. Informative References

   [I-D.bagnulo-ippm-new-registry-independent]
             Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and
             A. Morton, "A registry for commonly used metrics.
             Independent registries", draft-bagnulo-ippm-new-registry-
             independent-01 (work in progress), July 2013.

   [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-08 (work in progress), August 2014.

Authors' Addresses

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

   Phone: 34 91 6249500
   Email: marcelo@it.uc3m.es
   URI: http://www.it.uc3m.es

   Trevor Burbridge
   British Telecom
   Adastral Park, Martlesham Heath
   IPswitch
   ENGLAND

   Email: trevor.burbridge@bt.com

   Sam Crawford
   SamKnows

   Email: sam@samknows.com

Juergen Schoenwaelder
Jacobs University Bremen
Campus Ring 1
28759 Bremen
Germany

Email: j.schoenwaelder@jacobs-university.de

Vaibhav Bajpai
Jacobs University Bremen
Campus Ring 1
28759 Bremen
Germany

Email: v.bajpai@jacobs-university.de

LMAP Working Group   L. Deng
INTERNET-DRAFT   China Mobile
Intended Status: Informational   R. Huang
Expires: April 20, 2016   Huawei
                                                             S. Duan
                                                               CATR
                                                    October 19, 2015

                        Use-cases for Collaborative LMAP
                        draft-deng-lmap-collaboration-06

Abstract

   This document discusses the motivation and use-cases for
   collaborative LMAP practices, where multiple autonomous measurement
   systems collaborate together in performing large scale performance
   measurements to help with QoE enhancement by ICPs, network
   performance monitory to guide ISP/Regulator coordination between
   autonomous network domains and/or regulatory policies and cross-
   boundary troubleshooting for complaints from end consumers.

Status of this Memo

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

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as
   Internet-Drafts.

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

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

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

Table of Contents

1 Introduction

   With the rapid development of Internet technology and the increasing
   complexity of broadband network architecture, it is becoming
   difficult to do large scale network measurements due to the lack of
   the unified measurement system and cooperative protocols. Therefore,
   the Large-Scale Measurement of Broadband Performance (LMAP) working
   group is formed to standardize a large scale measurement system for
   the performance measurements of all kinds of broadband access
   methods.

   There are 3 types of entities proposed in the LMAP architecture: [I-
   D.ietf-lmap-framework]

   o Measurement Agents (MAs), implemented in network to perform
   measurement tasks;

   o Controller, responsible for creating and assigning the measurement
   tasks; and

   o Collector, in charge of collecting and storing measurement
   results.

   LMAP's current focus is to specify the information model, the
   associated data models, the control protocol for the secure
   communication between Controller and MA, and the report protocol for
   the secure communication between MA and Collector.

   On the other hand, for a large network, collaboration between
   multiple Controllers may also be needed for performing local
   measurement tasks, either because there is a practical limit on the
   number of MAs a single Controller can manage simultaneously for
   scalability considerations, because that a local task may involve
   multiple MAs that are speaking different languages (i.e. different
   control/report protocols), or because different organizations want to
   interconnect their measurement systems.

   Current LMAP protocols are designed under the following assumptions.

   o All the involved entities are under the control of a single
   organization.

   o An MA can only be controlled by a single controller at any given
   time.

o There is no communication between Controllers, between Collectors,
or between a Controller and a Collector.

However, cross-organization collaborations are increasingly common.
For example, accurate troubleshooting for mobile services usually
involves two or more organizations, and end-to-end performance
measurement may be conducted across multiple ISPs. How to utilize
LMAP practice to address these scenarios is still unsolved.

This document discusses the motivation and use-cases for
collaborative LMAP practices, where multiple autonomous measurement
systems collaborate together to help with QoE enhancement by ICPs,
network performance monitoring to guide planning for network
infrastructure and cross-boundary troubleshooting for SLA complaints
from end consumers, as well as performing regulatory supervision by
national regulators.

2   Terminology

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].

The following acronyms are used extensively in this document.

o ICP, Internet Content Provider.

o QoE, Quality of Experience.

o QoS, Quality of Service.

o ISP, Internet Service Provider, or shortly Operator.

o SLA, Service Level Agreement.

o UE, User Equipment.

o MAN, Metro Area Network.

o WAN, Wide Area Network.

The following definitions are borrowed from LMAP framework [I-D.ietf-
lmap-framework], and used to describe the corresponding entities
within a participating LMAP system.

o Controller: A function that provides a Measurement Agent with its
Instruction.

o Collector: A function that receives a Report from a Measurement Agent.

o Measurement Agent (MA): The function that receives Instruction Messages from a Controller and operates the Instruction by executing Measurement Tasks (using protocols outside the initial LMAP work scope and perhaps in concert with one or more other Measurement Agents or Measurement Peers) and (if part of the Instruction) by reporting Measurement Results to a Collector or Collectors.

o Measurement Method: The process for assessing the value of a Metric; the process of measuring some performance or reliability parameter associated with the transfer of traffic.

o Measurement Task: The action performed by a particular Measurement Agent that consists of the single assessment of a Metric through operation of a Measurement Method role at a particular time, with all of the role's Input Parameters set to specific values.

o Measurement Result: The output of a single Measurement Task (the value obtained for the parameter of interest or Metric).

o Metric: The quantity related to the performance and reliability of the network that we'd like to know the value of.

The following definitions are used in this document to describe corresponding entities for a collaborative performance measurement among multiple LMAP systems.

o Initiator, the instructor for collaborative Measurement Tasks, potentially on behalf of a regulator, a third party ICPs or an end consumer.

o Reporter, the reporting party that aggregates partial Measurements Reports from collaborative LMAP task participants and produces the ultimate report to the task Initiator.

o Region, a geographical area or administrative domain under the regulation of a single regulator.

o Domain, a collection of network devices and their interconnections under the operation of a single administrative entity.

3 Motivations for Collaborative LMAP

End-to-end performance measurement and trouble-shooting are important

for multiple parties, including: (1) Internet Service Providers, in
solving end user's QoE issues by better managing and optimizing their
networks, (2) Internet Content Providers, for enhance its service
logic and application design, (3) regulators in examining the status
of and guiding future regulation.

From ISP's perspective, the importance of supporting LMAP for its own
network construction and operation is without doubt. But taken into
account the potential impact of introducing third-party LMAP MAs into
key network entities, a sensible ISP would prefer to build its own
LMAP system based on MAs embedded into its local network devices.

It is hence expected that the majority of end-to-end performance
measurements will be conducted in a collaborative manner involving
multiple autonomous LMAP systems, for the following reasons:

On one hand, for the regulator, in order to stimulate network
development, it is necessary to have a clear picture of ISPs' peering
performance for interconnection points in addition to their own local
network construction. Considering the prohibitive cost of a unified
third-party deployment for LMAP MAs at various peering links among
ISPs for a large geographic area, it may be more practical to make
use of ISPs' autonomous LMAP systems for collaboration.

Let us take the example in China for instance. China's networks are
complex, with more than 31 provinces and 300 regions come to
hierarchical networks deployments. There are 3 ISP giants (CMCC,
CTCC, CUCC) in mainland China, managing nationwide hierarchical
networks, each is consisted of 3-4 national center points for
interconnecting on the top, more than 30 provincial backbone networks
in the middle, and more than 300 regions' local networks on the
bottom. In other words, the national regulator must know the network
status of the 3 networks in each region of a province, of a province,
and finally the whole country. It would be prohibitive for the
national regulator authority, MIIT to deploy its own dedicated probes
nationwide(900+).

Furthermore, regulators in different countries may want to
interconnect their measurement systems to perform cross-border
measurements.

On the other hand, for the ICP or user, it does not help much for
service optimization or trouble shooting if the end-to-end
performance measurement is conducted via a simple client-server model
while treating the network as a black box. In the meantime, for the
purpose of providing more value-added service to the ICPs as well as
subscribers, there is motive for an ISP to open its LMAP system to

some extent and collaborate with the ICP/user in understanding the
bottleneck and exploiting better network servicing for end-to-end
QoE.

In the following sections, more specific use-cases and derived
requirements of collaborative LMAP practices for end-to-end
performance measurement are presented.

4 Use-cases for Collaborative LMAP

As stated above, there are motivations from the regulator, ISP/ICP
and users to conduct collaborative measurements at the different
levels in order to know if the current network conditions meet the
expectations from the regulator policy, the ISP's resource provision
agreement or the ICP's service provision agreement. In particular,
the following usecases are identified.

4.1 Use-cases for Regulators

A regulator may want to monitor the current status and the future
deployment of network construction and operation of its region. In
order to promote network development, the regulator needs to monitor
the status of interconnection between different ISPs as well as the
overall network status.

4.1.1 within a regulator's own region

Understanding the current situation of its own region is necessary
for a regulator to form guiding policies for stimulating further
growth in high-speed networks. In order to get a clear picture of a
large geographic area, the regulator may choose to not deploy a
dedicated LMAP system on its own, while it's necessary to deploy a
large number of MAs. The regulator may achieve this goal by means of
the ISP's LMAP and the third-party LMAPs.

In that case, multiple organizations would simultaneously deploy
their dedicated MAs for private LMAP systems within their network
boundary in the same region, and by combining them together a
measurement system can mainly cover the whole region's network
infrastructure. Through collaboration, MAs from multiple
organizations can perform comprehensive measurement for the whole
regional network in great depth, which can reflect the network's
operational state.

4.1.2 peering performance between ISPs

Low performance of peering links between different ISPs not only has great impact on ICP services, but also on an access ISPs relying on transit ISPs for Internet connectivity. For example, a mobile operator lacking access to an Internet resource will have to pay interconnections to other operators. The regulator can formulate policies to promote information sharing between ISP networks and investigate the user QoE problem by understanding the interconnection performance. For the same reason, an ISP/ICP can also benefit from a more clear understanding of the performance of the interconnection.

For example, the data flow for a service request from a mobile terminal to an ICP first goes through the access ISP network and then into the Internet via a transit ISP network. Similarly, before entering the ICP's own private data-center, it may traverse another transit ISP network. As shown in Figure 1, the measurement can be implemented between ISP#1 MA and ISP#2 MA to understand the interconnection quality.

UE<=>access ISP<=>transit ISP #1<=>Internet<=>transit ISP #2<=>ICP

Figure 1 Cross-Domain data flow path

In a single administrative domain, there are also scenarios for collaborative measurement.

4.2 Use-cases for the ISP

4.2.1 measurements within a single domain

For one side, if the network scale is large enough, with many MAs, scalability of the Controller may become an issue [I-D.ooki-lmap-internet-measurement-system]. It would be a simple and scalable manner to construct an effective LMAP system by dividing the huge number of MAs into groups, and assign a Controller separately to manager each subset of MAs. The size of the MA groups are dependent on the number of MAs that a single Controller can manage at a time during the real deployment.

On the other hand, even the network scale is small, if there are many heterogeneous network devices as functioning MAs, the corresponding LMAP protocols/interface may be diverse. For example, browser built-in MAs can be conveniently implemented as HTTP clients, the CPE devices usually support TR.069 as their management protocol and network devices residing in the core network generally support and

runs SNMP protocol by default. In other words, different Controllers
speaking different LMAP protocols may be needed to respectively
manage different groups of MAs in the real deployment.

If a measurement task involves MAs that belong to different groups,
collaboration among corresponding Controllers is needed for
instructing the MAs with the task configuration and report
collection.

4.2.2 measurements for multi-domain ISP networks

   For a large ISP, it is common practice to divide its global network
   into several autonomous domains, each operated and managed by a
   regional branch. It is therefore, very likely that separate LMAP
   systems would be deployed into these autonomous domains, resulting in
   a call for collaborative measurement scenarios even within the same
   ISP's network.

   Take the case in China for instance, there are multiple nationwide
   ISP networks. Within these ISPs, relatively independent local
   branches, separated by physical territorial scope such as the
   province, operate their local network which has an autonomous domain
   or multiple autonomous domains. Each Provincial branch can deploy its
   own LMAP system to monitor its local network states.

4.3 Use-cases for the ICP

4.3.1 QoE-oriented performance enhancement

   New applications or updated applications with newly-added
   functions/features are being pushed to the end user every day, with
   an increasing requirement for constant performance optimization based
   on realistic network utilization resultant from application dynamics.
   It is important to understand the practical performance and impact of
   various network segments (e.g. access network, transit network and
   Internet) on the end-to-end traffic path. For the design,
   experimental and operational phases of a new feature/technology
   introduction to an application is also of great importance. However,
   it is expensive and non-economic for each ICP to build its own
   dedicated LMAP system into various ISPs' networks.

   At the same time, with the transition of ISPs' mindset from
   subscriber-centered charging for network access to ICP-centered

charging, ISPs are motivated to offer assistance to ICPs' exploration for better QoE through more efficient usage of network resources provisioned under the guidance of real-time performance measurements and optimization to accommodate application dynamics.

With ISPs' cooperation, various network segments are no longer hidden behind the black box to end-to-end performance measurements. By combining inputs from both its own end-based LMAP system with ISPs' measurement data, it is possible for an ICP to identify the bottleneck of service provision and develop corresponding enhancement via better guided technology introduction to the application as well as more targeted SLA negotiation with ISPs.

4.3.2 Trouble-shooting initiated by end consumers

With the growing influence of broadband access nowadays, more and more traditional ICPs are extending to the market of home gateways, as a result of the popularity of intelligent TVs and intelligent STBs. The services of end users in their home network are probably controlled by ICPs which may collaborate with the broadband access service providers to guarantee users the promised QoE. When malfunctions influencing user QoE occur in these types of services, it is necessary to have a mechanism with which the diagnostic measurement can be launched from the user side and identify the faulty party.

Generally the home gateway(such as a home WLAN router) is the border between the ISP network and the home network. The ISP network includes the access network, MAN and WAN. The home network includes home gateway, TV, STB, etc.

For a broadband access user who buys a third-party home gateway device, the typical service access path is shown in Figure 2. The home network between home gateway and UE is private and is not controlled by any ISP. However, the user may want to measure the link quality between the UE and the home gateway, the UE and the access ISP, or the UE to the ICP, separately. Thus in this scenario, it is difficult to deploy a single LMAP system which completely covers the whole path for accurate end-to-end QoE measurements and assists fault identification.

    UE <=>home net<=>home GW<=>access ISP<=>transit ISP<=>Internet<=>ICP

        Figure 2 Cross-Domain data traffic from home network to ICP

5 Derived Requirements

To make the requirements more clear, the following terms are defined:

LMAP domain: One LMAP domain is equal to one LMAP system specified in [i.d-ietf-lmap-framework], where all the MAs are controlled by a single controller.

This section presents derived requirements for LMAP protocols to enable the above collaborative use-cases across multiple LMAP domains. In particular:

* Current LMAP architecture MUST be extended to allow the MAs of a LMAP domain to accept the legal external measurement tasks initiated outside of the LMAP domain.

* When carrying out the outside measurement tasks, an LMAP domain MUST be able to coordinate the relevant controllers, MAs, and collectors of other LMAP domains for status updating or dynamical control.

* Current LMAP architecture MUST be extended to have a mechanism to gather and aggregate the measurement results from participating LMAP domains.

* An LMAP domain MUST be able to authenticate and authorize the measurement requests from outside of the LMAP domain.

* The extended mechanisms required above SHOULD NOT affect the current LMAP mechanisms in [i.d-ietf-lmap-framework]. If changes have to be made, they MUST be kept as small as possible.

6 Extension Discussions

In general, there are two basic approaches to extend the existing LMAP framework for the above requirements: the first is to add another layer of MA management and report collection for the additional information exchange; the other is to extend the existing controller/reporter's function and make one of the relevant controller/reporter to take the responsibility of collaborative task instruction/data aggregation.

6.1 Adding Another Layer of Management/Aggregation

In particular, two entities for the general coordination of cross-organization interactions for collaborative LMAP tasks are introduced: the Initiator and the Reporter, for cross-domain measurement task assignment and result aggregation, respectively. Three protocols for interactions for the newly-introduced entities

and existing LMAP entities are discussed too.

6.1.1 Initiator-Controller exchange for task instruction

   The globally trusted and verifiable Initiator instructs each
   participating LMAP Controller with corresponding Measurement Tasks to
   be performed within the LMAP system, indicating the corresponding
   Reporter, to whom the results of the Measurement Tasks are to be
   submitted. A globally unified identifier may be required for each
   collaborative Measurement Task.

6.1.2 Reporter-Collector exchange for data aggregation

   A Collector from each participating LMAP system interacts with the
   corresponding Reporter to report local measurement results.

6.1.3 Initiator-Reporter exchange for output instruction

   The Initiator also notifies the Reporter with instructions on how to
   create the final measurement report (e.g. data aggregation methods to
   be used) as well as the identities of the participating Controllers.

6.2 Extension over Existing Management/Aggregation Layer

   Another straightforward manner of extending the current LMAP
   framework to support collaborative measurements from multiple domains
   is to break the assumption that "any MA can only be controlled by a
   single Controller", and allow the MA within an LMAP domain to carry
   on the instructions from another Controller outside the domain,
   and/or report the measurement results to another outside Collector.

   Note that it is expected that such collaborative measurement
   instructions are not meant to change the ownership of the
   participating MA to its home LMAP domain.

   As long as there is not conflict of interest or competition of local
   resources at the MA, the outside measurement tasks (from an outside
   Controller outside the local LMAP domain) as well as all the inside
   measurement tasks (from the inside Controller in the local LMAP
   domain) can be carried on simultaneously.

   Otherwise, the MA may refer to static priority policies (e.g. the
   inside tasks have the top priority, etc.) or report to its local

Controller/a third party for conflict resolution and task adaptation.

7 Security Considerations

The security threats elaborated in [I-D.ietf-lmap-use-cases] also apply to collaborative LMAP scenarios.

It is assumed that the security issues within a participating LMAP system can be addressed by its local security mechanisms, as specified in [I-D.ietf-lmap-framework], and out of scope of this document.

Each participating LMAP system may have its own consideration and policy regarding its local network and/or subscriber private information. In performing collaborative task, it is still possible for a Collector to enforce local protection schemes, e.g. filtering algorithms, onto local measurement data before submission to the Reporter, hence providing protection to sensitive information for both the subscriber and the network operator.

It is important for a participating LMAP system to be able to authenticate the Initiator/outside-controller and the Reporter/outside-collector for a given collaborative Measurement Task, provide differentiated service provision according to its local policies (e.g. flexible authorization based on the Initiator's identity, the type of Measurement Task, Measurement Method, frequency, etc.), and protect itself from service abuse of malicious Initiators or information leakage to malicious Reporters.

A task/data verification scheme is needed for the Reporter to exclude un-authorized or non-intended Collectors from tampering the measurement report or blocking the Reporter/outside-collector from proper functioning with corrupted/forged/replayed local reports.

8 IANA Considerations

There is no IANA action in this document.

9 Acknowledgements

The authors would like to thank Charles Cook, Gregory Mirsky and Frode Sorensen for their valuable comments and input to this document.

10 References

10.1 Normative References

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

[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-11 (work in progress), February 2015.

[I-D.ietf-lmap-information-model] Burbridge, T., Eardley, P.,
            Bagnulo, M., and J. Schoenwaelder, "Information Model for
            Large-Scale Measurement Platforms (LMAP)", draft-ietf-
            lmap-information-model-03 (work in progress), January
            2015.

[I-D.ooki-lmap-internet-measurement-system] Ooki M., Kamei, S.,
            "Internet Measurement System", draft-ooki-lmap-internet-
            measurement-system-01(work in progress), December 2014.

[I-D.ietf-lmap-use-cases] Linsner M., Eardley, P., Burbridge, T.,
            Sorensen, F., "Large-Scale Broadband Measurement Use
            Cases", draft-ietf-lmap-use-cases-06(work in progress),
            Feburary, 2015

Authors' Addresses

    Lingli Deng
    China Mobile

    Email: denglingli@chinamobile.com

    Rachel Huang
    Huawei

    Email: rachel.huang@huawei.com

    Shihui Duan
    China Academy of Telecommunication Research of MIIT

    Email: duanshihui@catr.cn

Network Working Group   P. Eardley
Internet-Draft   BT
Intended status: Informational   A. Morton
Expires: October 31, 2015   AT&T Labs
                                                           M. Bagnulo
                                                                 UC3M
                                                         T. Burbridge
                                                                   BT
                                                            P. Aitken
                                                              Brocade
                                                            A. Akhter
                                                           Consultant
                                                       April 29, 2015

       A framework for Large-Scale Measurement of Broadband Performance (LMAP)
                    draft-ietf-lmap-framework-14

Abstract

   Measuring broadband service on a large scale requires a description
   of the logical architecture and standardisation of the key protocols
   that coordinate interactions between the components. The document
   presents an overall framework for large-scale measurements. It also
   defines terminology for LMAP (Large-Scale Measurement of Broadband
   Performance).

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 31, 2015.

Copyright Notice

Table of Contents

1. Introduction

   There is a desire to be able to coordinate the execution of broadband
   measurements and the collection of measurement results across a large
   scale set of Measurement Agents (MAs). These MAs could be software
   based agents on PCs, embedded agents in consumer devices (such as TVs
   or gaming consoles), embedded in service provider controlled devices
   such as set-top boxes and home gateways, or simply dedicated probes.
   MAs may also be embedded on a device that is part of an ISP's
   network, such as a DSLAM (Digital Subscriber Line Access
   Multiplexer), router, Carrier Grade NAT (Network Address Translator)
   or ISP Gateway. It is expected that a measurement system could
   easily encompass a few hundred thousand or even millions of such MAs.
   Such a scale presents unique problems in coordination, execution and
   measurement result collection. Several use cases have been proposed
   for large-scale measurements including:

   o Operators: to help plan their network and identify faults

   o Regulators: to benchmark several network operators and support
      public policy development

   Further details of the use cases can be found in
   [I-D.ietf-lmap-use-cases]. The LMAP framework should be useful for
   these, as well as other use cases, such as to help end users run
   diagnostic checks like a network speed test.

   The LMAP Framework has three basic elements: Measurement Agents,
   Controllers and Collectors.

   Measurement Agents (MAs) initiate the actual measurements, which are
   called Measurement Tasks in the LMAP terminology. In principle,
   there are no restrictions on the type of device in which the MA
   function resides.

   The Controller instructs one or more MAs and communicates the set of
   Measurement Tasks an MA should perform and when. For example it may
   instruct a MA at a home gateway: "Measure the 'UDP latency' with
   www.example.org; repeat every hour at xx.05". The Controller also
   manages a MA by instructing it how to report the Measurement Results,
   for example: "Report results once a day in a batch at 4am". We refer
   to these as the Measurement Schedule and Report Schedule.

   The Collector accepts Reports from the MAs with the Results from
   their Measurement Tasks. Therefore the MA is a device that gets
   Instructions from the Controller, initiates the Measurement Tasks,
   and reports to the Collector. The communications between these three

LMAP functions are structured according to a Control Protocol and a
Report Protocol.

The desirable features for a large-scale Measurement Systems we are
designing for are:

o Standardised - in terms of the Measurement Tasks that they
   perform, the components, the data models and protocols for
   transferring information between the components. Amongst other
   things, standardisation enables meaningful comparisons of
   measurements made of the same metric at different times and
   places, and provides the operator of a Measurement System with
   criteria for evaluation of the different solutions that can be
   used for various purposes including buying decisions (such as
   buying the various components from different vendors). Today's
   systems are proprietary in some or all of these aspects.

o Large-scale - [I-D.ietf-lmap-use-cases] envisages Measurement
   Agents in every home gateway and edge device such as set-top boxes
   and tablet computers, and located throughout the Internet as well
   [RFC7398]. It is expected that a Measurement System could easily
   encompass a few hundred thousand or even millions of Measurement
   Agents. Existing systems have up to a few thousand MAs (without
   judging how much further they could scale).

o Diversity - a Measurement System should handle Measurement Agents
   from different vendors, that are in wired and wireless networks,
   can execute different sorts of Measurement Task, are on devices
   with IPv4 or IPv6 addresses, and so on.

o Privacy Respecting - the protocols and procedures should respect
   the sensitive information of all those involved in measurements.

2. Outline of an LMAP-based measurement system

   In this section we provide an overview of the whole Measurement
   System. New LMAP-specific terms are capitalised; Section 3 provides
   a terminology section with a compilation of all the LMAP terms and
   their definition. Section 4 onwards considers the LMAP components in
   more detail.

   Other LMAP specifications will define an information model, the
   associated data models, and select/extend one or more protocols for
   the secure communication: firstly, a Control Protocol, from a
   Controller to instruct Measurement Agents what performance metrics to
   measure, when to measure them, how/when to report the measurement
   results to a Collector; secondly, a Report Protocol, for a
   Measurement Agent to report the results to the Collector.

The Figure below shows the main components of a Measurement System,
and the interactions of those components. Some of the components are
outside the scope of initial LMAP work.

The MA performs Measurement Tasks. One possibility is that the MA is
observes existing traffic. Another possibility is for the MA to
generate (or receive) traffic specially created for the purpose and
measure some metric associated with its transfer. The
Figure includes both possibilities (in practice, it may be more usual
for a MA to do one) whilst Section 6.4 shows some examples of
possible arrangements of the components.

The MAs are pieces of code that can be executed in specialised
hardware (hardware probe) or on a general-purpose device (like a PC
or mobile phone). A device with a Measurement Agent may have
multiple physical interfaces (Wi-Fi, Ethernet, DSL (Digital
Subscriber Line); and non-physical interfaces such as PPPoE (Point-
to-Point Protocol over Ethernet) or IPsec) and the Measurement Tasks
may specify any one of these.

The Controller manages a MA through use of the Control Protocol,
which transfers the Instruction to the MA. This describes the
Measurement Tasks the MA should perform and when. For example the
Controller may instruct a MA at a home gateway: "Count the number of
TCP SYN packets observed in a 1 minute interval; repeat every hour at
xx.05 + Unif[0,180] seconds". The Measurement Schedule determines
when the Measurement Tasks are executed. The Controller also manages
a MA by instructing it how to report the Measurement Results, for
example: "Report results once a day in a batch at 4am + Unif[0,180]
seconds; if the end user is active then delay the report 5 minutes".
The Report Schedule determines when the Reports are uploaded to the
Collector. The Measurement Schedule and Report Schedule can define
one-off (non-recurring) actions ("Do measurement now", "Report as
soon as possible"), as well as recurring ones.

The Collector accepts a Report from a MA with the Measurement Results
from its Measurement Tasks. It then provides the Results to a
repository (see below).

A Measurement Method defines how to measure a Metric of interest. It
is very useful to standardise Measurement Methods, so that it is
meaningful to compare measurements of the same Metric made at
different times and places. It is also useful to define a registry
for commonly-used Metrics [I-D.ietf-ippm-metric-registry] so that a
Metric with its associated Measurement Method can be referred to
simply by its identifier in the registry. The registry will
hopefully be referenced by other standards organisations. The

Measurement Methods may be defined by the IETF, locally, or by some
other standards body.

Broadly speaking there are two types of Measurement Method. In both
types a Measurement Agent measures a particular Observed Traffic
Flow. It may involve a single MA simply observing existing traffic -
for example, the Measurement Agent could count bytes or calculate the
average loss for a particular flow. On the other hand, a Measurement
Method may involve multiple network entities, which perform different
roles. For example, a "ping" Measurement Method, to measure the
round trip delay , would consist of an MA sending an ICMP (Internet
Control Message Protocol) ECHO request to a responder in the
Internet. In LMAP terms, the responder is termed a Measurement Peer
(MP), meaning that it helps the MA but is not managed by the
Controller. Other Measurement Methods involve a second MA, with the
Controller instructing the MAs in a coordinated manner. Traffic
generated specifically as part of the Measurement Method is termed
Measurement Traffic; in the ping example, it is the ICMP ECHO
Requests and Replies. The protocols used for the Measurement Traffic
are out of the scope of initial LMAP work, and fall within the scope
of other IETF WGs such as IPPM (IP Performance Metrics).

A Measurement Task is the action performed by a particular MA at a
particular time, as the specific instance of its role in a
Measurement Method. LMAP is mainly concerned with Measurement Tasks,
for instance in terms of its Information Model and Protocols.

For Measurement Results to be truly comparable, as might be required
by a regulator, not only do the same Measurement Methods need to be
used to assess Metrics, but also the set of Measurement Tasks should
follow a similar Measurement Schedule and be of similar number. The
details of such a characterisation plan are beyond the scope of work
in IETF although certainly facilitated by IETF's work.

Both control and report messages are transferred over a secure
Channel. A Control Channel is between the Controller and a MA; the
Control Protocol delivers Instruction Messages to the MA and
Capabilities, Failure and Logging Information in the reverse
direction. A Report Channel is between a MA and Collector, and the
Report Protocol delivers Reports to the Collector.

Finally we introduce several components that are outside the scope of
initial LMAP work and will be provided through existing protocols or
applications. They affect how the Measurement System uses the
Measurement Results and how it decides what set of Measurement Tasks
to perform. As shown in the Figure, these components are: the
bootstrapper, Subscriber parameter database, data analysis tools, and
Results repository.

The MA needs to be bootstrapped with initial details about its
Controller, including authentication credentials. The LMAP work
considers the bootstrap process, since it affects the Information
Model. However, LMAP does not define a bootstrap protocol, since it
is likely to be technology specific and could be defined by the
Broadband Forum, CableLabs or IEEE depending on the device. Possible
protocols are SNMP (Simple Network Management Protocol), NETCONF
(Network Configuration Protocol) or (for Home Gateways) CPE WAN
Management Protocol (CWMP) from the Auto Configuration Server (ACS)
(as specified in TR-069 [TR-069]).

A Subscriber parameter database contains information about the line,
such as the customer's broadband contract (perhaps 2, 40 or 80Mb/s),
the line technology (DSL or fibre), the time zone where the MA is
located, and the type of home gateway and MA. These parameters are
already gathered and stored by existing operations systems. They may
affect the choice of what Measurement Tasks to run and how to
interpret the Measurement Results. For example, a download test
suitable for a line with an 80Mb/s contract may overwhelm a 2Mb/s
line.

A Results repository records all Measurement Results in an equivalent
form, for example an SQL (Structured Query Language) database, so
that they can easily be accessed by the data analysis tools.

The data analysis tools receive the results from the Collector or via
the Results repository. They might visualise the data or identify
which component or link is likely to be the cause of a fault or
degradation. This information could help the Controller decide what
follow-up Measurement Task to perform in order to diagnose a fault.
The data analysis tools also need to understand the Subscriber's
service information, for example the broadband contract.

```
 +--------+      +----------+                 +----------+   ^
 |End user|      |          |    Observed     | End user |   |
 |        |<-----|----------|---traffic--->|  |          |   |
 |        |      |          |    flow          |          |   |
 |        |      |          |                 |          |  Non-LMAP
 |        |      |          |   Measurement   |          |   Scope
 |        |      |          |   <--traffic--->|          |   |
 +--------+      |          |                 +----------+   |
 ............|...........|................................V
 <MP>        |Measurement|                  <MP>          ^
             |Agent:     |                                |
             |LMAP       |                                |
 +---------->|interface  |                                |
 |           +----------+                                 LMAP
 |            ^   |     Report                            Scope
 |      Instruction |   | (over Report Channel)           |
 |      (over Control |  +--------------------+           |
 |        Channel)  |  |                    |           |
 |           |      |  |                    |           |
 |           |      |  |                    v           |
 |           |      |  +-----------+   +-----------+    |
 |           | Controller |        |   | Collector |    |
 |           | +-----------+        +-----------+    v
 |           |  ^       ^                  |        ^
 |           |  |       |                  |        |
 |           |  |   +--------+             |        |
 |           |  |   |        |             v        |
 +-----------+  +----------+ +--------+ +----------+ |
 |Bootstrapper| |Subscriber|--->| data  |<---| Results  |  Non-
 +-----------+  |parameter | |analysis|    |repository|  LMAP
                |database  | | tools  |    +----------+  Scope
                +----------+ +--------+                   |
                                                          |
                                                          v
```
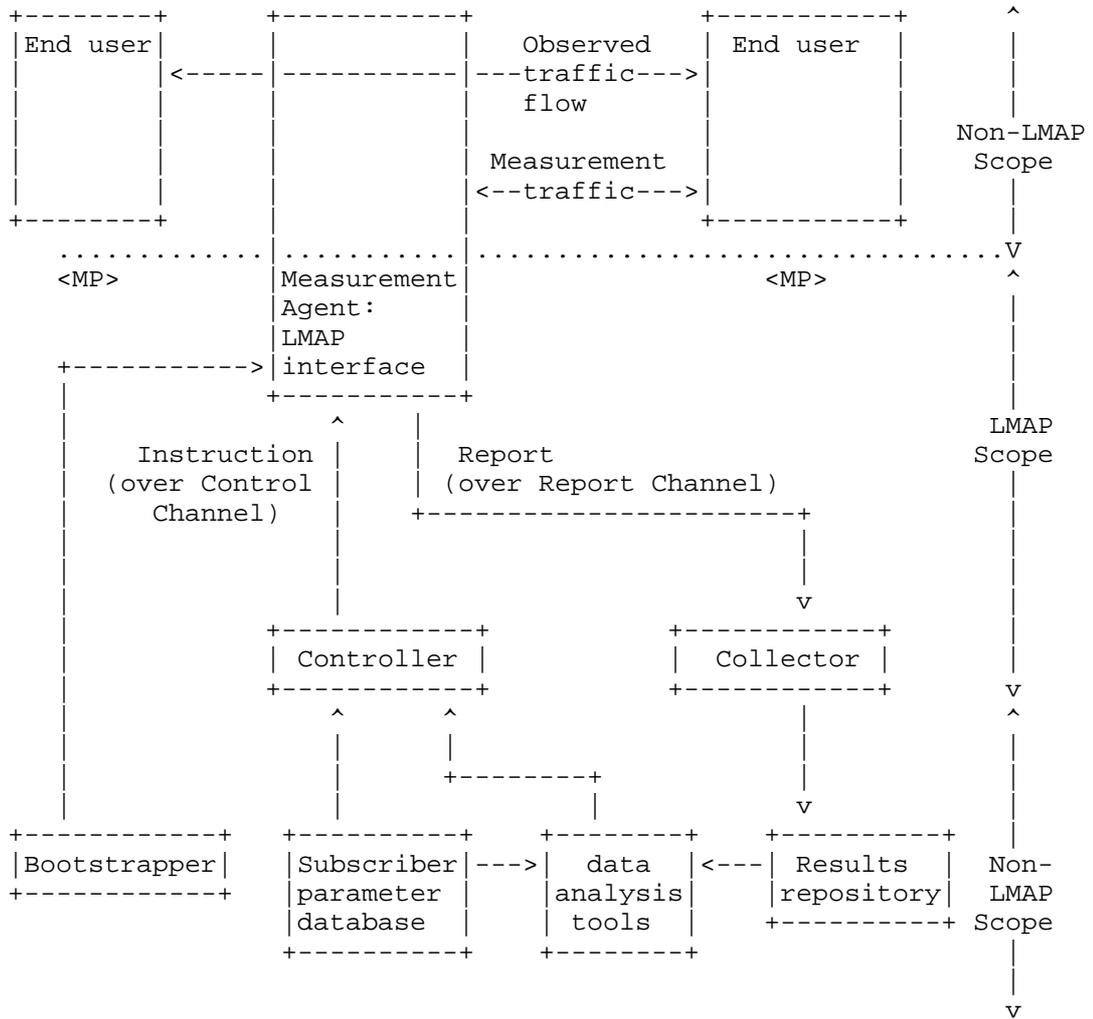
   Schematic of main elements of an LMAP-based Measurement System
   (showing the elements in and out of the scope of initial LMAP work)

3. Terminology

   This section defines terminology for LMAP. Please note that defined
   terms are capitalized.

   Bootstrap: A process that integrates a Measurement Agent into a
   Measurement System.

Capabilities: Information about the performance measurement
capabilities of the MA, in particular the Measurement Method roles
and measurement protocol roles that it can perform, and the device
hosting the MA, for example its interface type and speed, but not
dynamic information.

Channel: A bi-directional logical connection that is defined by a
specific Controller and MA, or Collector and MA, plus associated
security.

Collector: A function that receives a Report from a Measurement
Agent.

Configuration: A process for informing the MA about its MA-ID,
(optional) Group-ID and Control Channel.

Controller: A function that provides a Measurement Agent with its
Instruction.

Control Channel: A Channel between a Controller and a MA over which
Instruction Messages and Capabilities, Failure and Logging
Information are sent.

Control Protocol: The protocol delivering Instruction(s) from a
Controller to a Measurement Agent. It also delivers Capabilities,
Failure and Logging Information from the Measurement Agent to the
Controller. It can also be used to update the MA's Configuration.
It runs over the Control Channel.

Cycle-ID: A tag that is sent by the Controller in an Instruction and
echoed by the MA in its Report. The same Cycle-ID is used by several
MAs that use the same Measurement Method for a Metric with the same
Input Parameters. Hence the Cycle-ID allows the Collector to easily
identify Measurement Results that should be comparable.

Data Model: The implementation of an Information Model in a
particular data modelling language [RFC3444].

Environmental Constraint: A parameter that is measured as part of the
Measurement Task, its value determining whether the rest of the
Measurement Task proceeds.

Failure Information: Information about the MA's failure to action or
execute an Instruction, whether concerning Measurement Tasks or
Reporting.

Group-ID: An identifier of a group of MAs.

Information Model: The protocol-neutral definition of the semantics of the Instructions, the Report, the status of the different elements of the Measurement System as well of the events in the system [RFC3444].

Input Parameter: A parameter whose value is left open by the Metric and its Measurement Method and is set to a specific value in a Measurement Task. Altering the value of an Input Parameter does not change the fundamental nature of the Measurement Task.

Instruction: The description of Measurement Tasks for a MA to perform and the details of the Report for it to send. It is the collective description of the Measurement Task configurations, the configuration of the Measurement Schedules, the configuration of the Report Channel(s), the configuration of Report Schedule(s), and the details of any suppression.

Instruction Message: The message that carries an Instruction from a Controller to a Measurement Agent.

Logging Information: Information about the operation of the Measurement Agent, which may be useful for debugging.

Measurement Agent (MA): The function that receives Instruction Messages from a Controller and operates the Instruction by executing Measurement Tasks (using protocols outside the initial LMAP work scope and perhaps in concert with one or more other Measurement Agents or Measurement Peers) and (if part of the Instruction) by reporting Measurement Results to a Collector or Collectors.

Measurement Agent Identifier (MA-ID): a UUID [RFC4122] that identifies a particular MA and is configured as part of the Bootstrapping process.

Measurement Method: The process for assessing the value of a Metric; the process of measuring some performance or reliability parameter associated with the transfer of traffic.

Measurement Peer (MP): The function that assists a Measurement Agent with Measurement Tasks and does not have an interface to the Controller or Collector.

Measurement Result: The output of a single Measurement Task (the value obtained for the parameter of interest or Metric).

Measurement Schedule: The schedule for performing Measurement Tasks.

   Measurement System: The set of LMAP-defined and related components
   that are operated by a single organisation, for the purpose of
   measuring performance aspects of the network.

   Measurement Task: The action performed by a particular Measurement
   Agent that consists of the single assessment of a Metric through
   operation of a Measurement Method role at a particular time, with all
   of the role's Input Parameters set to specific values.

   Measurement Traffic: the packet(s) generated by some types of
   Measurement Method that involve measuring some parameter associated
   with the transfer of the packet(s).

   Metric: The quantity related to the performance and reliability of
   the network that we'd like to know the value of.

   Observed Traffic Flow: In RFC 7011, a Traffic Flow (or Flow) is
   defined as a set of packets or frames passing an Observation Point in
   the network during a certain time interval. All packets belonging to
   a particular Flow have a set of common properties, such as packet
   header fields, characteristics, and treatments. A Flow measured by
   the LMAP system is termed an Observed Traffic Flow. Its properties
   are summarized and tabulated in Measurement Results (as opposed to
   raw capture and export).

   Report: The set of Measurement Results and other associated
   information (as defined by the Instruction). The Report is sent by a
   Measurement Agent to a Collector.

   Report Channel: A Channel between a Collector and a MA over which
   Report messages are sent.

   Report Protocol: The protocol delivering Report(s) from a Measurement
   Agent to a Collector. It runs over the Report Channel.

   Report Schedule: the schedule for sending Reports to a Collector.

   Subscriber: An entity (associated with one or more users) that is
   engaged in a subscription with a service provider.

   Suppression: the temporary cessation of Measurement Tasks.

4. Constraints

   The LMAP framework makes some important assumptions, which constrain
   the scope of the initial LMAP work.

4.1. The measurement system is under the direction of a single
      organisation

   In the LMAP framework, the Measurement System is under the direction
   of a single organisation that is responsible for any impact that its
   measurements have on a user's quality of experience and privacy.
   Clear responsibility is critical given that a misbehaving large-scale
   Measurement System could potentially harm user experience, user
   privacy and network security.

   However, the components of an LMAP Measurement System can be deployed
   in administrative domains that are not owned by the measuring
   organisation. Thus, the system of functions deployed by a single
   organisation constitutes a single LMAP domain which may span
   ownership or other administrative boundaries.

4.2. Each MA may only have a single Controller at any point in time

   A MA is instructed by one Controller and is in one Measurement
   System. The constraint avoids different Controllers giving a MA
   conflicting instructions and so means that the MA does not have to
   manage contention between multiple Measurement (or Report) Schedules.
   This simplifies the design of MAs (critical for a large-scale
   infrastructure) and allows a Measurement Schedule to be tested on
   specific types of MA before deployment to ensure that the end user
   experience is not impacted (due to CPU, memory or broadband-product
   constraints). However, a Measurement System may have several
   Controllers.

5. Protocol Model

   A protocol model [RFC4101] presents an architectural model for how
   the protocol operates and needs to answer three basic questions:

   1. What problem is the protocol trying to address?

   2. What messages are being transmitted and what do they mean?

   3. What are the important, but unobvious, features of the protocol?

   An LMAP system goes through the following phases:

   o a Bootstrapping process before the MA can take part in the other
      three phases.

   o a Control Protocol, which delivers Instruction Messages from a
      Controller to a MA (amongst other things).

o the actual Measurement Tasks, which measure some performance or
   reliability parameter(s) associated with the transfer of packets.

o a Report Protocol, which delivers Reports containing the
   Measurement Results from a MA to a Collector.

The diagrams show the various LMAP messages and uses the following
convention:

o (optional): indicated by round brackets

o [potentially repeated]: indicated by square brackets

The protocol model is closely related to the Information Model
[I-D.ietf-lmap-information-model], which is the abstract definition
of the information carried by the protocol. (If there is any
difference between this document and the Information Model, the
latter is definitive, since it is on the standards track.) The
purpose of both is to provide a protocol and device independent view,
which can be implemented via specific protocols. LMAP defines a
specific Control Protocol and Report Protocol, but others could be
defined by other standards bodies or be proprietary. However it is
important that they all implement the same Information Model and
protocol model, in order to ease the definition, operation and
interoperability of large-scale Measurement Systems.

5.1. Bootstrapping process

The primary purpose of bootstrapping is to enable a MA to be
integrated into a Measurement System. The MA retrieves information
about itself (like its identity in the Measurement System) and about
the Controller, the Controller learns information about the MA, and
they learn about security information to communicate (such as
certificates and credentials).

Whilst this memo considers the bootstrapping process, it is beyond
the scope of initial LMAP work to define a bootstrap mechanism, as it
depends on the type of device and access.

As a result of the bootstrapping process the MA learns information
with the following aims ([I-D.ietf-lmap-information-model] defines
the consequent list of information elements):

o its identifier, either its MA-ID or a device identifier such as
   one of its MAC or both.

o (optionally) a Group-ID. A Group-ID would be shared by several
   MAs and could be useful for privacy reasons. For instance,

reporting the Group-ID and not the MA-ID could hinder tracking of a mobile device

o the Control Channel, which is defined by:

* the address which identifies the Control Channel, such as the Controller's FQDN (Fully Qualified Domain Name) [RFC1035])

* security information (for example to enable the MA to decrypt the Instruction Message and encrypt messages sent to the Controller)

The details of the bootstrapping process are device /access specific. For example, the information could be in the firmware, manually configured or transferred via a protocol like TR-069 [TR-069]. There may be a multi-stage process where the MA contacts a 'hard-coded' address, which replies with the bootstrapping information.

The MA must learn its MA-ID before getting an Instruction, either during Bootstrapping or via Configuration (Section 5.2.1).

5.2. Control Protocol

The primary purpose of the Control Protocol is to allow the Controller to configure a Measurement Agent with an Instruction about what Measurement Tasks to do, when to do them, and how to report the Measurement Results (Section 5.2.2). The Measurement Agent then acts on the Instruction autonomously. The Control Protocol also enables the MA to inform the Controller about its Capabilities and any Failure and Logging Information (Section 5.2.2). Finally, the Control Protocol allows the Controller to update the MA's Configuration.

5.2.1. Configuration

Configuration allows the Controller to update the MA about some or all of the information that it obtained during the bootstrapping process: the MA-ID, the (optional) Group-ID and the Control Channel. The Measurement System might use Configuration for several reasons. For example, the bootstrapping process could 'hard code' the MA with details of an initial Controller, and then the initial Controller could configure the MA with details about the Controller that sends Instruction Messages. (Note that a MA only has one Control Channel, and so is associated with only one Controller, at any moment.)

Note that an implementation may choose to combine Configuration information and an Instruction Message into a single message.

```
   +----------------+                               +-------------+
   |                |                               | Measurement |
   |   Controller   |===============================| Agent       |
   +----------------+                               +-------------+
```

   Configuration information: ->
   (MA-ID),
   (Group-ID),
   (Control Channel)

                                           <-   Response(details)
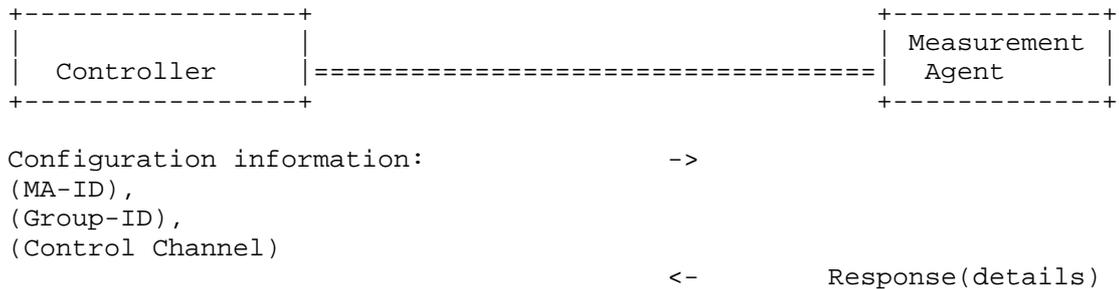
5.2.2. Instruction

   The Instruction is the description of the Measurement Tasks for a
   Measurement Agent to do and the details of the Measurement Reports
   for it to send. In order to update the Instruction the Controller
   uses the Control Protocol to send an Instruction Message over the
   Control Channel.

```
   +----------------+                               +-------------+
   |                |                               | Measurement |
   |   Controller   |===============================| Agent       |
   +----------------+                               +-------------+
```

   Instruction: ->
   [(Measurement Task configuration
      URI of Metric(
     [Input Parameter],
     (Role)
     (interface),
     (Cycle-ID)
     (measurement point)),
    (Report Channel),
    (Schedule),
    (Suppression information)]

                                           <-   Response(details)
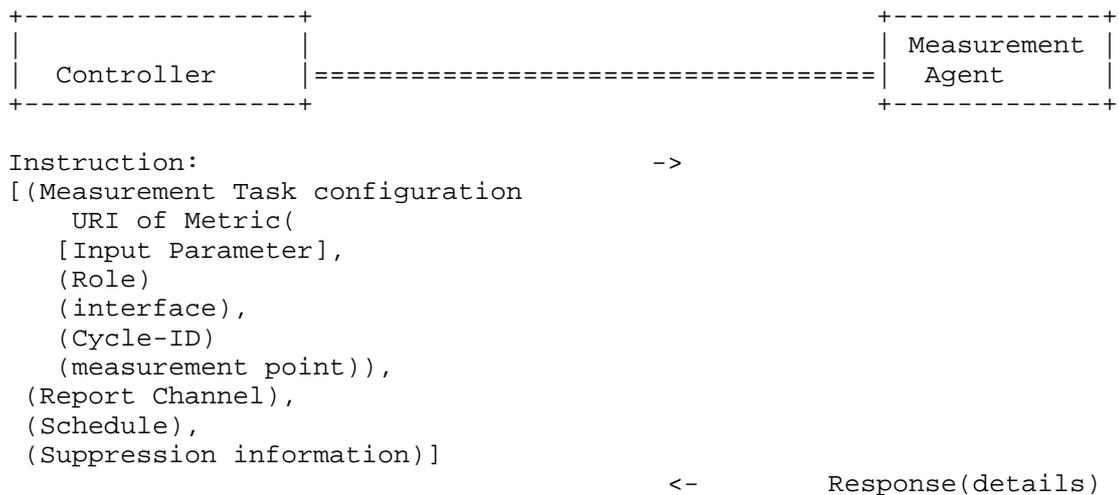
   The Instruction defines information with the following aims
   ([I-D.ietf-lmap-information-model] defines the consequent list of
   information elements):

   o the Measurement Task configurations, each of which needs:

      * the Metric, specified as a URI to a registry entry; it includes
         the specification of a Measurement Method. The registry could

        be defined by a standards organisation or locally by the
        operator of the Measurement System. Note that, at the time of
        writing, the IETF works on such a registry specification
        [I-D.ietf-ippm-metric-registry].

     * the Measurement Method role. For some Measurement Methods,
        different parties play different roles; for example (see
        Section 6.4) an iperf sender and receiver. Each Metric and its
        associated Measurement Method will describe all measurement
        roles involved in the process.

     * a boolean flag (suppress or do-not-suppress) indicating if such
        a Measurement Task is impacted by a Suppression message (see
        Section 5.2.2.1). Thus, the flag is an Input Parameter.

     * any Input Parameters that need to be set for the Metric and the
        Measurement Method. For example, the address of a Measurement
        Peer (or other Measurement Agent) that may be involved in a
        Measurement Task , or traffic filters associated with the
        Observed Traffic Flow.

     * if the device with the MA has multiple interfaces, then the
        interface to use (if not defined, then the default interface is
        used).

     * optionally, a Cycle-ID.

     * optionally, the measurement point designation [RFC7398] of the
        MA and, if applicable, of the MP or other MA. This can be
        useful for reporting.

   o configuration of the Schedules, each of which needs:

     * the timing of when the Measurement Tasks are to be performed,
        or the Measurement Reports are to be sent. Possible types of
        timing are periodic, calendar-based periodic, one-off immediate
        and one-off at a future time

   o configuration of the Report Channel(s), each of which needs:

     * the address of the Collector, for instance its URL

     * security for this Report Channel, for example the X.509
        certificate

   o Suppression information, if any (see Section 5.2.1.1)

A single Instruction Message may contain some or all of the above
parts. The finest level of granularity possible in an Instruction
Message is determined by the implementation and operation of the
Control Protocol. For example, a single Instruction Message may add
or update an individual Measurement Schedule - or it may only update
the complete set of Measurement Schedules; a single Instruction
Message may update both Measurement Schedules and Measurement Task
configurations - or only one at a time; and so on. However,
Suppression information always replaces (rather than adds to) any
previous Suppression information.

The MA informs the Controller that it has successfully understood the
Instruction Message, or that it cannot action the Instruction - for
example, if it doesn't include a parameter that is mandatory for the
requested Metric and Measurement Method, or it is missing details of
the target Collector.

The Instruction Message instructs the MA; the Control Protocol does
not allow the MA to negotiate, as this would add complexity to the
MA, Controller and Control Protocol for little benefit.

5.2.2.1. Suppression

The Instruction may include Suppression information. The main
motivation for Suppression is to enable the Measurement System to
eliminate Measurement Traffic, because there is some unexpected
network issue for example. There may be other circumstances when
Suppression is useful, for example to eliminate inessential Reporting
traffic (even if there is no Measurement Traffic).

The Suppression information may include any of the following optional
fields:

o a set of Measurement Tasks to suppress; the others are not
   suppressed. For example, this could be useful if a particular
   Measurement Task is overloading a Measurement Peer with
   Measurement Traffic.

o a set of Measurement Schedules to suppress; the others are not
   suppressed. For example, suppose the Measurement System has
   defined two Schedules, one with the most critical Measurement
   Tasks and the other with less critical ones that create a lot of
   Measurement Traffic, then it may only want to suppress the second.

o a set of Reporting Schedules to suppress; the others are not
   suppressed. This can be particularly useful in the case of a
   Measurement Method that doesn't generate Measurement Traffic; it

may need to continue observing traffic flows but temporarily
suppress Reports due to the network footprint of the Reports.

o if all the previous fields are included then the MA suppresses the
   union - in other words, it suppresses the set of Measurement
   Tasks, the set of Measurement Schedules, and the set of Reporting
   Schedules.

o if the Suppression information includes neither a set of
   Measurement Tasks nor a set of Measurement Schedules, then the MA
   does not begin new Measurement Tasks that have the boolean flag
   set to "suppress"; however, the MA does begin new Measurement
   Tasks that have the flag set to "do-not-suppress".

o a start time, at which suppression begins. If absent, then
   Suppression begins immediately.

o an end time, at which suppression ends. If absent, then
   Suppression continues until the MA receives an un-Suppress
   message.

o a demand that the MA immediately ends on-going Measurement Task(s)
   that are tagged for suppression. (Most likely it is appropriate
   to delete the associated partial Measurement Result(s).) This
   could be useful in the case of a network emergency so that the
   operator can eliminate all inessential traffic as rapidly as
   possible. If absent, the MA completes on-going Measurement Tasks.

An un-Suppress message instructs the MA no longer to suppress,
meaning that the MA once again begins new Measurement Tasks,
according to its Measurement Schedule.

Note that Suppression is not intended to permanently stop a
Measurement Task (instead, the Controller should send a new
Measurement Schedule), nor to permanently disable a MA (instead, some
kind of management action is suggested).

```
   +----------------+                          +------------+
   |                |                          | Measurement |
   |   Controller   |==========================| Agent      |
   +----------------+                          +------------+
```
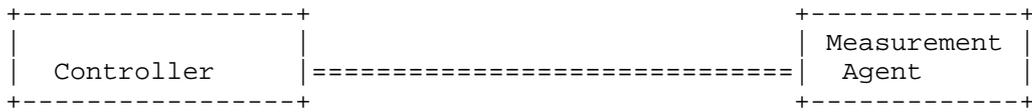
```
   Suppress:
   [(Measurement Task),                ->
    (Measurement Schedule),
    [start time],
    [end time],
    [on-going suppressed?]]

   Un-suppress                         ->
```

5.2.3. Capabilities, Failure and Logging Information

   The Control Protocol also enables the MA to inform the Controller
   about various information, such as its Capabilities and any Failures.
   It is also possible to use a device-specific mechanism which is
   beyond the scope of the initial LMAP work.

   Capabilities are information about the MA that the Controller needs
   to know in order to correctly instruct the MA, such as:

   o the Measurement Method (roles) that the MA supports

   o the measurement protocol types and roles that the MA supports

   o the interfaces that the MA has

   o the version of the MA

   o the version of the hardware, firmware or software of the device
      with the MA

   o its Instruction (this could be useful if the Controller thinks
      something has gone wrong, and wants to check what Instruction the
      MA is using)

   o but not dynamic information like the currently unused CPU, memory
      or battery life of the device with the MA.

   Failure Information concerns why the MA has been unable to execute a
   Measurement Task or deliver a Report, for example:

   o the Measurement Task failed to run properly because the MA
      (unexpectedly) has no spare CPU cycles

o the MA failed to record the Measurement Results because it
   (unexpectedly) is out of spare memory

o a Report failed to deliver Measurement Results because the
   Collector (unexpectedly) is not responding

o but not if a Measurement Task correctly doesn't start. For
   example, the first step of some Measurement Methods is for the MA
   to check there is no cross-traffic.

Logging Information concerns how the MA is operating and may help
debugging, for example:

o the last time the MA ran a Measurement Task

o the last time the MA sent a Measurement Report

o the last time the MA received an Instruction Message

o whether the MA is currently Suppressing Measurement Tasks

Capabilities, Failure and Logging Information are sent by the MA,
either in response to a request from the Controller (for example, if
the Controller forgets what the MA can do or otherwise wants to
resynchronize what it knows about the MA), or on its own initiative
(for example when the MA first communicates with a Controller or if
it becomes capable of a new Measurement Method). Another example of
the latter case is if the device with the MA re-boots, then the MA
should notify its Controller in case its Instruction needs to be
updated; to avoid a "mass calling event" after a widespread power
restoration affecting many MAs, it is sensible for an MA to pause for
a random delay, perhaps in the range of one minute or so.

```
+-----------------+                              +------------+
|                 |                              | Measurement |
|   Controller    |==============================| Agent      |
+-----------------+                              +------------+

(Instruction:
  [(Request Capabilities),
    (Request Failure Information),
    (Request Logging Information),
    (Request Instruction)])          ->
                                     <-       (Capabilities),
                                          (Failure Information),
                                          (Logging Information),
                                              (Instruction)
```

5.3. Operation of Measurement Tasks

   This LMAP framework is neutral to what the actual Measurement Task
   is. It does not define Metrics and Measurement Methods, these are
   defined elsewhere.

   The MA carries out the Measurement Tasks as instructed, unless it
   gets an updated Instruction. The MA acts autonomously, in terms of
   operation of the Measurement Tasks and reporting of the Results; it
   doesn't do a 'safety check' with the Controller to ask whether it
   should still continue with the requested Measurement Tasks.

   The MA may operate Measurement Tasks sequentially or in parallel (see
   Section 5.3.2).

5.3.1. Starting and Stopping Measurement Tasks

   This LMAP framework does not define a generic start and stop process,
   since the correct approach depends on the particular Measurement
   Task; the details are defined as part of each Measurement Method.
   This section provides some general hints. The MA does not inform the
   Controller about Measurement Tasks starting and stopping.

   Before beginning a Measurement Task the MA may want to run a pre-
   check. (The pre-check could be defined as a separate, preceding Task
   or as the first part of a larger Task.)

   For Measurement Tasks that observe existing traffic, action could
   include:

   o checking that there is traffic of interest;

   o checking that the device with the MA has enough resources to
      execute the Measurement Task reliably. Note that the designer of
      the Measurement System should ensure that the device's
      capabilities are normally sufficient to comfortably operate the
      Measurement Tasks.

   For Measurement Tasks that generate Measurement Traffic, a pre-check
   could include:

   o the MA checking that there is no cross-traffic. In other words, a
      check that the end-user isn't already sending traffic;

   o the MA checking with the Measurement Peer (or other Measurement
      Agent) involved in the Measurement Task that it can handle a new
      Measurement Task. For example, the Measurement Peer may already
      be handling many Measurement Tasks with other MAs;

o sending traffic that probes the path to check it isn't overloaded;

o checking that the device with the MA has enough resources to
   execute the Measurement Task reliably.

It is possible that similar checks continue during the Measurement
Task, especially one that is long-running and/or creates a lot of
Measurement Traffic, and might lead to it being abandoned whilst in-
progress. A Measurement Task could also be abandoned in response to
a "suppress" message (see Section 5.2.1). Action could include:

o For 'upload' tests, the MA not sending traffic

o For 'download' tests, the MA closing the TCP connection or sending
   a TWAMP (Two-Way Active Measurement Protocol) Stop control message
   [RFC5357].

The Controller may want a MA to run the same Measurement Task
indefinitely (for example, "run the 'upload speed' Measurement Task
once an hour until further notice"). To avoid the MA generating
traffic forever after a Controller has permanently failed (or
communications with the Controller have failed), the MA can be
configured with a time limit; if the MA doesn't hear from the
Controller for this length of time, then it stops operating
Measurement Tasks.

5.3.2. Overlapping Measurement Tasks

It is possible that a MA starts a new Measurement Task before another
Measurement Task has completed. This may be intentional (the way
that the Measurement System has designed the Measurement Schedules),
but it could also be unintentional - for instance, if a Measurement
Task has a 'wait for X' step which pauses for an unexpectedly long
time. This document makes no assumptions about the impact of one
Measurement Task on another.

The operator of the Measurement System can handle (or not)
overlapping Measurement Tasks in any way they choose - it is a policy
or implementation issue and not the concern of LMAP. Some possible
approaches are: to configure the MA not to begin the second
Measurement Task; to start the second Measurement Task as usual; for
the action to be an Input Parameter of the Measurement Task; and so
on.

It may be important to include in the Measurement Report the fact
that the Measurement Task overlapped with another.

5.4. Report Protocol

   The primary purpose of the Report Protocol is to allow a Measurement
   Agent to report its Measurement Results to a Collector, along with
   the context in which they were obtained.
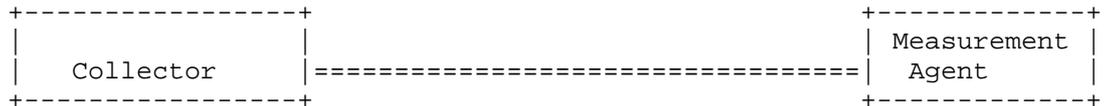
```
+----------------+                               +------------+
|                |                               | Measurement |
|   Collector    |===============================| Agent      |
+----------------+                               +------------+

                              <-      Report:
                                          [MA-ID &/or Group-ID],
                                           [Measurement Result],
                                   [details of Measurement Task],
                                                     [Cycle-ID]
ACK                           ->
```

   The Report contains:

   o the MA-ID or a Group-ID (to anonymise results)

   o the actual Measurement Results, including the time they were
      measured. In general the time is simply the MA's best estimate
      and there is no guarantee on the accuracy or granularity of the
      information. It is possible that some specific analysis of a
      particular Measurement Method's Results will impose timing
      requirements.

   o the details of the Measurement Task (to avoid the Collector having
      to ask the Controller for this information later). For example,
      the interface used for the measurements.

   o the Cycle-ID, if one was included in the Instruction.

   o perhaps the Subscriber's service parameters (see Section 5.4.1).

   o the measurement point designation of the MA and, if applicable,
      the MP or other MA, if the information was included in the
      Instruction. This numbering system is defined in [RFC7398] and
      allows a Measurement Report to describe abstractly the path
      measured (for example, "from a MA at a home gateway to a MA at a
      DSLAM"). Also, the MA can anonymise results by including
      measurement point designations instead of IP addresses
      (Section 8.6.2).

The MA sends Reports as defined by the Instruction. It is possible
that the Instruction tells the MA to report the same Results to more
than one Collector, or to report a different subset of Results to
different Collectors. It is also possible that a Measurement Task
may create two (or more) Measurement Results, which could be reported
differently (for example, one Result could be reported periodically,
whilst the second Result could be an alarm that is created as soon as
the measured value of the Metric crosses a threshold and that is
reported immediately).

Optionally, a Report is not sent when there are no Measurement
Results.

In the initial LMAP Information Model and Report Protocol, for
simplicity we assume that all Measurement Results are reported as-is,
but allow extensibility so that a Measurement System (or perhaps a
second phase of LMAP) could allow a MA to:

o label, or perhaps not include, Measurement Results impacted by,
   for instance, cross-traffic or a Measurement Peer (or other
   Measurement Agent) being busy

o label Measurement Results obtained by a Measurement Task that
   overlapped with another

o not report the Measurement Results if the MA believes that they
   are invalid

o detail when Suppression started and ended

As discussed in Section 6.1, data analysis of the results should
carefully consider potential bias from any Measurement Results that
are not reported, or from Measurement Results that are reported but
may be invalid.

5.4.1. Reporting of Subscriber's service parameters

The Subscriber's service parameters are information about his/her
broadband contract, line rate and so on. Such information is likely
to be needed to help analyse the Measurement Results, for example to
help decide whether the measured download speed is reasonable.

The information could be transferred directly from the Subscriber
parameter database to the data analysis tools. If the subscriber's
service parameters are available to the MAs, they could be reported
with the Measurement Results in the Report Protocol. How (and if)
the MA knows such information is likely to depend on the device type.

The MA could either include the information in a Measurement Report or separately.

5.5. Operation of LMAP over the underlying packet transfer mechanism

The above sections have described LMAP's protocol model. Other specifications will define the actual Control and Report Protocols, possibly operating over an existing protocol, such as REST-style HTTP(S). It is also possible that a different choice is made for the Control and Report Protocols, for example NETCONF-YANG [RFC6241] and IPFIX (Internet Protocol Flow Information Export) [RFC7011] respectively.

From an LMAP perspective, the Controller needs to know that the MA has received the Instruction Message, or at least that it needs to be re-sent as it may have failed to be delivered. Similarly the MA needs to know about the delivery of Capabilities and Failure information to the Controller and Reports to the Collector. How this is done depends on the design of the Control and Report Protocols and the underlying packet transfer mechanism.

For the Control Protocol, the underlying packet transfer mechanism could be:

o a 'push' protocol (that is, from the Controller to the MA)

o a multicast protocol (from the Controller to a group of MAs)

o a 'pull' protocol. The MA periodically checks with Controller if the Instruction has changed and pulls a new Instruction if necessary. A pull protocol seems attractive for a MA behind a NAT or firewall (as is typical for a MA on an end-user's device), so that it can initiate the communications. It also seems attractive for a MA on a mobile device, where the Controller might not know how to reach the MA. A pull mechanism is likely to require the MA to be configured with how frequently it should check in with the Controller, and perhaps what it should do if the Controller is unreachable after a certain number of attempts.

o a hybrid protocol. In addition to a pull protocol, the Controller can also push an alert to the MA that it should immediately pull a new Instruction.

For the Report Protocol, the underlying packet transfer mechanism could be:

o a 'push' protocol (that is, from the MA to the Collector)

   o perhaps supplemented by the ability for the Collector to 'pull'
      Measurement Results from a MA.

5.6. Items beyond the scope of the initial LMAP work

   There are several potential interactions between LMAP elements that
   are beyond the scope of the initial LMAP work:

   1. It does not define a coordination process between MAs. Whilst a
       Measurement System may define coordinated Measurement Schedules
       across its various MAs, there is no direct coordination between
       MAs.

   2. It does not define interactions between the Collector and
       Controller. It is quite likely that there will be such
       interactions, optionally intermediated by the data analysis
       tools. For example, if there is an "interesting" Measurement
       Result then the Measurement System may want to trigger extra
       Measurement Tasks that explore the potential cause in more
       detail; or if the Collector unexpectedly does not hear from a MA,
       then the Measurement System may want to trigger the Controller to
       send a fresh Instruction Message to the MA.

   3. It does not define coordination between different Measurement
       Systems. For example, it does not define the interaction of a MA
       in one Measurement System with a Controller or Collector in a
       different Measurement System. Whilst it is likely that the
       Control and Report Protocols could be re-used or adapted for this
       scenario, any form of coordination between different
       organisations involves difficult commercial and technical issues
       and so, given the novelty of large-scale measurement efforts, any
       form of inter-organisation coordination is outside the scope of
       the initial LMAP work. Note that a single MA is instructed by a
       single Controller and is only in one Measurement System.

       * An interesting scenario is where a home contains two
          independent MAs, for example one controlled by a regulator and
          one controlled by an ISP. Then the Measurement Traffic of one
          MA is treated by the other MA just like any other end-user
          traffic.

   4. It does not consider how to prevent a malicious party "gaming the
       system". For example, where a regulator is running a Measurement
       System in order to benchmark operators, a malicious operator
       could try to identify the broadband lines that the regulator was
       measuring and prioritise that traffic. It is assumed this is a
       policy issue and would be dealt with through a code of conduct
       for instance.

   5. It does not define how to analyse Measurement Results, including
       how to interpret missing Results.

   6. It does not specifically define a end-user-controlled Measurement
       System, see sub-section 5.6.1.

5.6.1. End-user-controlled measurement system

   This framework concentrates on the cases where an ISP or a regulator
   runs the Measurement System. However, we expect that LMAP
   functionality will also be used in the context of an end-user-
   controlled Measurement System. There are at least two ways this
   could happen (they have various pros and cons):

   1. an end-user could somehow request the ISP- (or regulator-) run
       Measurement System to test his/her line. The ISP (or regulator)
       Controller would then send an Instruction to the MA in the usual
       LMAP way.

   2. an end-user could deploy their own Measurement System, with their
       own MA, Controller and Collector. For example, the user could
       implement all three functions onto the same end-user-owned end
       device, perhaps by downloading the functions from the ISP or
       regulator. Then the LMAP Control and Report Protocols do not
       need to be used, but using LMAP's Information Model would still
       be beneficial. A Measurement Peer (or other MA involved in a
       Measurement Task) could be in the home gateway or outside the
       home network; in the latter case the Measurement Peer is highly
       likely to be run by a different organisation, which raises extra
       privacy considerations.

   In both cases there will be some way for the end-user to initiate the
   Measurement Task(s). The mechanism is outside the scope of the
   initial LMAP work, but could include the user clicking a button on a
   GUI or sending a text message. Presumably the user will also be able
   to see the Measurement Results, perhaps summarised on a webpage. It
   is suggested that these interfaces conform to the LMAP guidance on
   privacy in Section 8.

6. Deployment considerations

6.1. Controller and the measurement system

   The Controller should understand both the MA's LMAP Capabilities (for
   instance what Metrics and Measurement Methods it can perform) and
   about the MA's other capabilities like processing power and memory.
   This allows the Controller to make sure that the Measurement Schedule

of Measurement Tasks and the Reporting Schedule are sensible for each
MA that it instructs.

An Instruction is likely to include several Measurement Tasks.
Typically these run at different times, but it is also possible for
them to run at the same time. Some Tasks may be compatible, in that
they do not affect each other's Results, whilst with others great
care would need to be taken. Some Tasks may be complementary. For
example, one Task may be followed by a traceroute Task to the same
destination address, in order to learn the network path that was
measured.

The Controller should ensure that the Measurement Tasks do not have
an adverse effect on the end user. Tasks, especially those that
generate a substantial amount of Measurement Traffic, will often
include a pre-check that the user isn't already sending traffic
(Section 5.3). Another consideration is whether Measurement Traffic
will impact a Subscriber's bill or traffic cap.

A Measurement System may have multiple Controllers (but note the
overriding principle that a single MA is instructed by a single
Controller at any point in time (Section 4.2)). For example, there
could be different Controllers for different types of MA (home
gateways, tablets) or locations (Ipswich, Edinburgh, Paris), for load
balancing or to cope with failure of one Controller.

The measurement system also needs to consider carefully how to
interpret missing Results. The correct interpretation depends on why
the Results are missing (perhaps related to measurement suppression
or delayed Report submission), and potentially on the specifics of
the Measurement Task and Measurement Schedule. For example, the set
of packets represented by a Flow may be empty; that is, an Observed
Traffic Flow may represent zero or more packets. The Flow would
still be reported according to schedule.

6.2. Measurement Agent

The MA should be cautious about resuming Measurement Tasks if it re-
boots or has been off-line for some time, as its Instruction may be
stale. In the former case it also needs to ensure that its clock has
re-set correctly, so that it interprets the Schedule correctly.

If the MA runs out of storage space for Measurement Results or can't
contact the Controller, then the appropriate action is specific to
the device and Measurement System.

The Measurement Agent could take a number of forms: a dedicated
probe, software on a PC, embedded into an appliance, or even embedded

into a gateway. A single site (home, branch office etc.) that is
participating in a measurement could make use of one or multiple
Measurement Agents or Measurement Peers in a single measurement.

The Measurement Agent could be deployed in a variety of locations.
Not all deployment locations are available to every kind of
Measurement Agent. There are also a variety of limitations and
trade-offs depending on the final placement. The next sections
outline some of the locations a Measurement Agent may be deployed.
This is not an exhaustive list and combinations may also apply.

6.2.1. Measurement Agent on a networked device

A MA may be embedded on a device that is directly connected to the
network, such as a MA on a smartphone. Other examples include a MA
downloaded and installed on a subscriber's laptop computer or tablet
when the network service is provided on wired or other wireless radio
technologies, such as Wi-Fi.

6.2.2. Measurement Agent embedded in site gateway

A Measurement Agent embedded with the site gateway, for example a
home router or the edge router of a branch office in a managed
service environment, is one of better places the Measurement Agent
could be deployed. All site-to-ISP traffic would traverse through
the gateway. So, Measurement Methods that measure user traffic could
easily be performed. Similarly, due to this user traffic visibility,
a Measurement Method that generates Measurement Traffic could ensure
it does not compete with user traffic. Generally NAT and firewall
services are built into the gateway, allowing the Measurement Agent
the option to offer its Controller-facing management interface
outside of the NAT/firewall. This placement of the management
interface allows the Controller to unilaterally contact the
Measurement Agent for instructions. However, a Measurement Agent on
a site gateway (whether end-user service-provider owned) will
generally not be directly available for over the top providers, the
regulator, end users or enterprises.

6.2.3. Measurement Agent embedded behind site NAT /firewall

The Measurement Agent could also be embedded behind a NAT, a
firewall, or both. In this case the Controller may not be able to
unilaterally contact the Measurement Agent unless either static port
forwarding or firewall pin holing is configured. Configuring port
forwarding could use protocols such as PCP [RFC6887], TR-069 [TR-069]
or UPnP [UPnP]. To open a pin hole in the firewall, the Measurement
Agent could send keepalives towards the Controller (and perhaps use
these also as a network reachability test).

6.2.4. Multi-homed Measurement Agent

   If the device with the Measurement Agent is single homed then there
   is no confusion about what interface to measure. Similarly, if the
   MA is at the gateway and the gateway only has a single WAN-side and a
   single LAN-side interface, there is little confusion - for
   Measurement Methods that generate Measurement Traffic, the location
   of the other MA or Measurement Peer determines whether the WAN or LAN
   is measured.

   However, the device with the Measurement Agent may be multi-homed.
   For example, a home or campus may be connected to multiple broadband
   ISPs, such as a wired and wireless broadband provider, perhaps for
   redundancy or load- sharing. It may also be helpful to think of dual
   stack IPv4 and IPv6 broadband devices as multi-homed. More
   generally, Section 3.2 of [RFC7368] describes dual-stack and multi-
   homing topologies that might be encountered in a home network,
   [RFC6419] provides the current practices of multi-interfaces hosts,
   and the Multiple Interfaces (mif) working group covers cases where
   hosts are either directly attached to multiple networks (physical or
   virtual) or indirectly (multiple default routers, etc.). In these
   cases, there needs to be clarity on which network connectivity option
   is being measured.

   One possibility is to have a Measurement Agent per interface. Then
   the Controller's choice of MA determines which interface is measured.
   However, if a MA can measure any of the interfaces, then the
   Controller defines in the Instruction which interface the MA should
   use for a Measurement Task; if the choice of interface is not defined
   then the MA uses the default one. Explicit definition is preferred
   if the Measurement System wants to measure the performance of a
   particular network, whereas using the default is better if the
   Measurement System wants to include the impact of the MA's interface
   selection algorithm. In any case, the Measurement Result should
   include the network that was measured.

6.2.5. Measurement Agent embedded in ISP network

   A MA may be embedded on a device that is part of an ISP's network,
   such as a router or switch. Usually the network devices with an
   embedded MA will be strategically located, such as a Carrier Grade
   NAT or ISP Gateway. [RFC7398] gives many examples where a MA might
   be located within a network to provide an intermediate measurement
   point on the end-to-end path. Other examples include a network
   device whose primary role is to host MA functions and the necessary
   measurement protocol.

6.3. Measurement Peer

   A Measurement Peer participates in some Measurement Methods. It may
   have specific functionality to enable it to participate in a
   particular Measurement Method. On the other hand, other Measurement
   Methods may require no special functionality. For example if the
   Measurement Agent sends a ping to example.com then the server at
   example.com plays the role of a Measurement Peer; or if the MA
   monitors existing traffic, then the existing end points are
   Measurement Peers.

   A device may participate in some Measurement Methods as a Measurement
   Agent and in others as a Measurement Peer.

   Measurement Schedules should account for limited resources in a
   Measurement Peer when instructing a MA to execute measurements with a
   Measurement Peer. In some measurement protocols, such as [RFC4656]
   and [RFC5357], the Measurement Peer can reject a measurement session
   or refuse a control connection prior to setting-up a measurement
   session and so protect itself from resource exhaustion. This is a
   valuable capability because the MP may be used by more than one
   organisation.

6.4. Deployment examples

   In this section we describe some deployment scenarios that are
   feasible within the LMAP framework defined in this document.

   A very simple example of a Measurement Peer (MP) is a web server that
   the MA is downloading a web page from (such as www.example.com) in
   order to perform a speed test. The web server is a MP and from its
   perspective, the MA is just another client; the MP doesn't have a
   specific function for assisting measurements. This is described in
   the figure below.

```
                                                          ^
        +------------------+  Web Traffic +----------------+ non-LMAP
        |     Web Client   |<------------>|   Web Server   | Scope
        |                  |              +----------------+   |
    ...|..................|.......................................V...
        |MA:LMAP interface |                 <MP:>            ^
        +------------------+                                  |
              ^        |                                      |
    Instruction |      | Report                               |
              |        +----------------+                     |
              |        |                |                     |
              |                         v            LMAP     |
        +-----------+            +-----------+        Scope    |
        | Controller |           | Collector |         |
        +-----------+            +-----------+         V
```
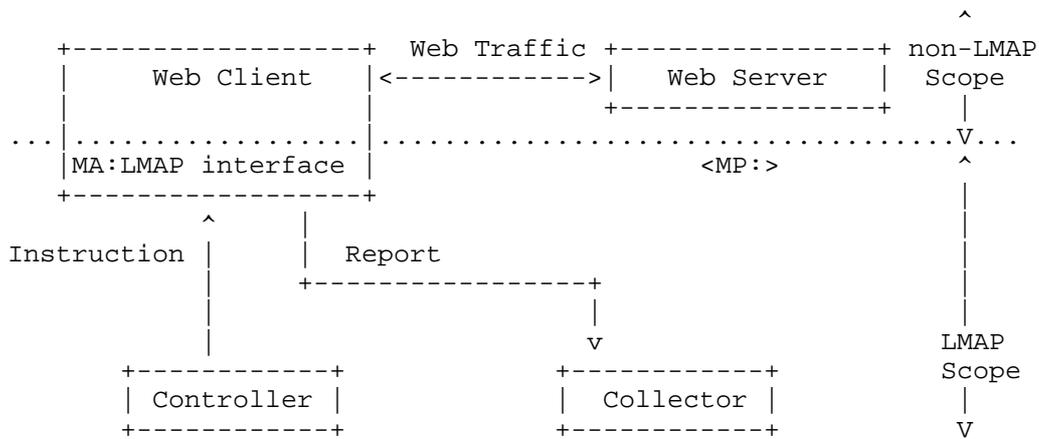
Schematic of LMAP-based Measurement System,
with Web server as Measurement Peer

Another case that is slightly different than this would be the one of
a TWAMP-responder. This is also a MP, with a helper function, the
TWAMP server, which is specially deployed to assist the MAs that
perform TWAMP tests. Another example is with a ping server, as
described in Section 2.

A further example is the case of a traceroute like measurement. In
this case, for each packet sent, the router where the TTL expires is
performing the MP function. So for a given Measurement Task, there
is one MA involved and several MPs, one per hop.

In the figure below we depict the case of an OWAMP (One-Way Active
Measurement Protocol) responder acting as an MP. In this case, the
helper function in addition reports results back to the MA. So it
has both a data plane and control interface with the MA.

```
+-----------------+    OWAMP   +----------------+   ^
| OWAMP           |<--control--->|              |   |
| control-client  |-test-traffic>| OWAMP server & | non-LMAP
| fetch-client &  |<----fetch----| session-rec'ver| Scope
| session-sender  |              |                |   |
|                 |              +----------------+   |
...|.................|..............................................v...
   |MA:LMAP interface |                  <MP:>              ^
   +-----------------+                                      |
        ^        |                                          |
Instruction |    | Report                                   |
        |    +----------------+                             |
        |    |                |                             |
        |    |                v                          LMAP
    +-----------+        +-----------+                   Scope
    | Controller |       | Collector |                     |
    +-----------+        +-----------+                     v
```
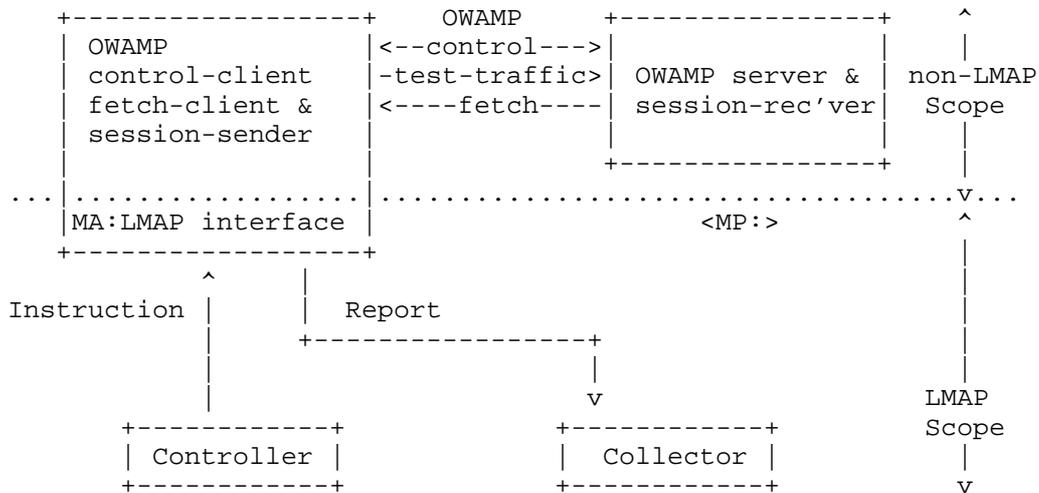
Schematic of LMAP-based Measurement System,
with OWAMP server as Measurement Peer

However, it is also possible to use two Measurement Agents when
performing one way Measurement Tasks, as described in the figure
below. Both MAs are instructed by the Controller: MA-1 to send the
traffic and MA-2 to measure the received traffic and send Reports to
the Collector. Note that the Measurement Task at MA-2 can listen for
traffic from MA-1 and respond multiple times without having to be
rescheduled.

```
+----------------+            +----------------+   ^
|                |            |                |   | non-LMAP
| iperf -u sender|-UDP traffic->| iperf -u recvr | Scope
|                |            |                |   v
...|................|...............|.................|........
   |  MA-1:         |            |  MA-2:         |   ^
   | LMAP interface |            | LMAP interface |   |
   +----------------+            +----------------+   |
        ^            Instruction{Report} |    ^      |
Instruction |                       |  |  | Report   |
{task,      |    +------------------+  |  |          |
 schedule}  |    |                     |  |          |
        |    |                         v  LMAP
    +-----------+        +-----------+   Scope
    | Controller |       | Collector |     |
    +-----------+        +-----------+     v
```
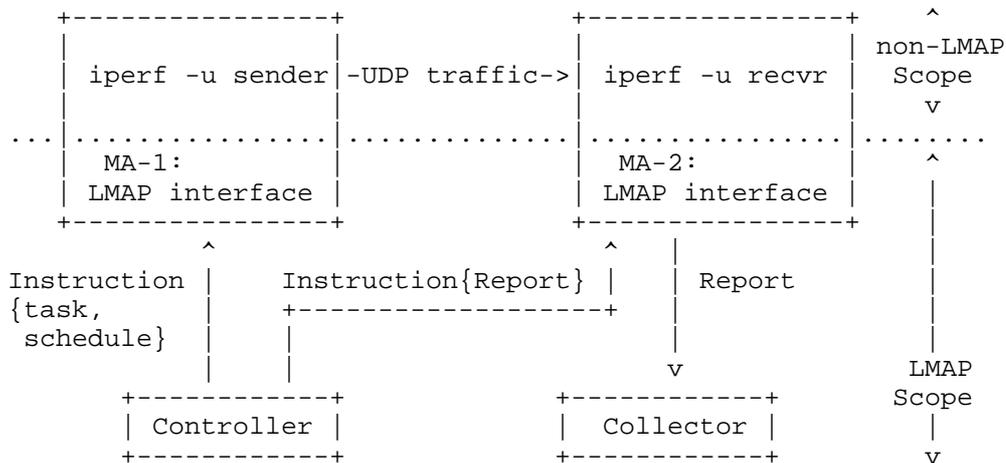
Schematic of LMAP-based Measurement System, with two
Measurement Agents cooperating to measure UDP traffic

Next, we consider Measurement Methods that meter the Observed Traffic
Flow. Traffic generated in one point in the network flowing towards
a given destination and the traffic is observed in some point along
the path. One way to implement this is that the endpoints generating
and receiving the traffic are not instructed by the Controller; hence
they are MPs. The MA is located along the path with a monitor
function that measures the traffic. The MA is instructed by the
Controller to monitor that particular traffic and to send the Report
to the Collector. It is depicted in the figure below.

```
+--------+  +------------------+   +--------+   ^
|End user|  |     Monitor      | Observed |End user|   |
|        |<--|------------------|--traffic-->|        |  non-LMAP
|        |  |                  |   flow   |        |   |  Scope
+--------+  |                  |          +--------+   |
        ...|..................|...........................v..
           |MA:LMAP interface |            <MP:>         ^
           +------------------+                          |
                   ^     |                               |
       Instruction |     | Report                        |
                   |     +----------------+              |
                   |     |                |              |
                   |     |                v              LMAP
           +-----------+          +-----------+   Scope
           | Controller |          | Collector |   |
           +-----------+          +-----------+   v
```
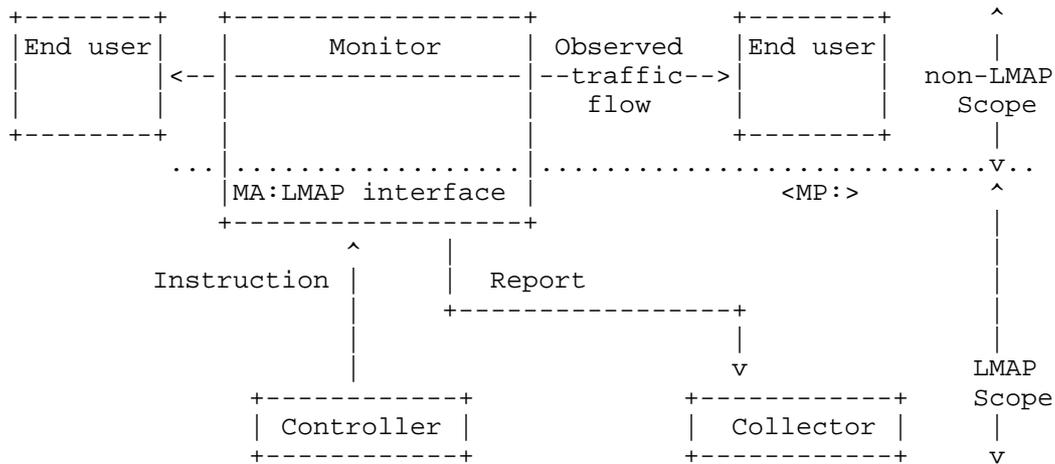
Schematic of LMAP-based Measurement System,
with a Measurement Agent monitoring traffic

7. Security considerations

   The security of the LMAP framework should protect the interests of
   the measurement operator(s), the network user(s) and other actors who
   could be impacted by a compromised measurement deployment. The
   Measurement System must secure the various components of the system
   from unauthorised access or corruption. Much of the general advice
   contained in section 6 of [RFC4656] is applicable here.

   The process to upgrade the firmware in an MA is outside the scope of
   the initial LMAP work, just as is the protocol to bootstrap the MAs.
   However, systems which provide remote upgrade must secure authorised
   access and integrity of the process.

   We assume that each Measurement Agent (MA) will receive its
   Instructions from a single organisation, which operates the
   Controller. These Instructions must be authenticated (to ensure that
   they come from the trusted Controller), checked for integrity (to

ensure no-one has tampered with them) and not vulnerable to replay
attacks. If a malicious party can gain control of the MA they can
use it to launch DoS attacks at targets, create a platform for
pervasive monitoring [RFC7258], reduce the end user's quality of
experience and corrupt the Measurement Results that are reported to
the Collector. By altering the Measurement Tasks and/or the address
that Results are reported to, they can also compromise the
confidentiality of the network user and the MA environment (such as
information about the location of devices or their traffic). The
Instruction Messages also need to be encrypted to maintain
confidentiality, as the information might be useful to an attacker.

Reporting by the MA must be encrypted to maintain confidentiality, so
that only the authorised Collector can decrypt the results, to
prevent the leakage of confidential or private information.
Reporting must also be authenticated (to ensure that it comes from a
trusted MA and that the MA reports to a genuine Collector) and not
vulnerable to tampering (which can be ensured through integrity and
replay checks). It must not be possible to fool a MA into injecting
falsified data and the results must also be held and processed
securely after collection and analysis. See section 8.5.2 below for
additional considerations on stored data compromise, and section 8.6
on potential mitigations for compromise.

Since Collectors will be contacted repeatedly by MAs using the
Collection Protocol to convey their recent results, a successful
attack to exhaust the communication resources would prevent a
critical operation: reporting. Therefore, all LMAP Collectors should
implement technical mechanisms to:

o limit the number of reporting connections from a single MA
   (simultaneous, and connections per unit time).

o limit the transmission rate from a single MA.

o limit the memory/storage consumed by a single MA's reports.

o efficiently reject reporting connections from unknown sources.

o separate resources if multiple authentication strengths are used,
   where the resources should be separated according to each class of
   strength.

A corrupted MA could report falsified information to the Collector.
Whether this can be effectively mitigated depends on the platform on
which the MA is deployed, but where the MA is deployed on a customer-
controlled device then the reported data is to some degree inherently
untrustworthy. Further, a sophisticated party could distort some

Measurement Methods, perhaps by dropping or delaying packets for
example. This suggests that the network operator should be cautious
about relying on Measurement Results for action such as refunding
fees if a service level agreement is not met.

As part of the protocol design, it will be decided how LMAP operates
over the underlying protocol (Section 5.5). The choice raises
various security issues, such as how to operate through a NAT and how
to protect the Controller and Collector from denial of service
attacks.

The security mechanisms described above may not be strictly necessary
if the network's design ensures the LMAP components and their
communications are already secured, for example potentially if they
are all part of an ISP's dedicated management network.

Finally, there are three other issues related to security: privacy
(considered in Section 8 below), availability and 'gaming the
system'. While the loss of some MAs may not be considered critical,
the unavailability of the Collector could mean that valuable business
data or data critical to a regulatory process is lost. Similarly,
the unavailability of a Controller could mean that the MAs do not
operate a correct Measurement Schedule.

A malicious party could "game the system". For example, where a
regulator is running a Measurement System in order to benchmark
operators, an operator could try to identify the broadband lines that
the regulator was measuring and prioritise that traffic. Normally,
this potential issue is handled by a code of conduct. It is outside
the scope of the initial LMAP work to consider the issue.

8. Privacy considerations

The LMAP work considers privacy as a core requirement and will ensure
that by default the Control and Report Protocols operate in a
privacy-sensitive manner and that privacy features are well-defined.

This section provides a set of privacy considerations for LMAP. This
section benefits greatly from the timely publication of [RFC6973].
Privacy and security (Section 7) are related. In some jurisdictions
privacy is called data protection.

We begin with a set of assumptions related to protecting the
sensitive information of individuals and organisations participating
in LMAP-orchestrated measurement and data collection.

8.1. Categories of entities with information of interest

   LMAP protocols need to protect the sensitive information of the
   following entities, including individuals and organisations who
   participate in measurement and collection of results.

   o Individual Internet users: Persons who utilise Internet access
      services for communications tasks, according to the terms of
      service of a service agreement. Such persons may be a service
      Subscriber, or have been given permission by the Subscriber to use
      the service.

   o Internet service providers: Organisations who offer Internet
      access service subscriptions, and thus have access to sensitive
      information of individuals who choose to use the service. These
      organisations desire to protect their Subscribers and their own
      sensitive information which may be stored in the process of
      performing Measurement Tasks and collecting Results.

   o Regulators: Public authorities responsible for exercising
      supervision of the electronic communications sector, and which may
      have access to sensitive information of individuals who
      participate in a measurement campaign. Similarly, regulators
      desire to protect the participants and their own sensitive
      information.

   o Other LMAP system operators: Organisations who operate Measurement
      Systems or participate in measurements in some way.

   Although privacy is a protection extended to individuals, we discuss
   data protection by ISPs and other LMAP system operators in this
   section. These organisations have sensitive information involved in
   the LMAP system, and many of the same dangers and mitigations are
   applicable. Further, the ISPs store information on their Subscribers
   beyond that used in the LMAP system (for instance billing
   information), and there should be a benefit in considering all the
   needs and potential solutions coherently.

8.2. Examples of sensitive information

   This section gives examples of sensitive information which may be
   measured or stored in a Measurement System, and which is to be kept
   private by default in the LMAP core protocols.

   Examples of Subscriber or authorised Internet user sensitive
   information:

o Sub-IP layer addresses and names (MAC address, base station ID, SSID)

o IP address in use

o Personal Identification (real name)

o Location (street address, city)

o Subscribed service parameters

o Contents of traffic (activity, DNS queries, destinations, equipment types, account info for other services, etc.)

o Status as a study volunteer and Schedule of Measurement Tasks

Examples of Internet Service Provider sensitive information:

o Measurement device identification (equipment ID and IP address)

o Measurement Instructions (choice of measurements)

o Measurement Results (some may be shared, others may be private)

o Measurement Schedule (exact times)

o Network topology (locations, connectivity, redundancy)

o Subscriber billing information, and any of the above Subscriber information known to the provider.

o Authentication credentials (such as certificates)

Other organisations will have some combination of the lists above. The LMAP system would not typically expose all of the information above, but could expose a combination of items which could be correlated with other pieces collected by an attacker (as discussed in the section on Threats below).

8.3. Different privacy issues raised by different sorts of Measurement Methods

Measurement Methods raise different privacy issues depending on whether they measure traffic created specifically for that purpose, or whether they measure user traffic.

Measurement Tasks conducted on user traffic store sensitive information, however briefly this storage may be. We note that some

authorities make a distinction on time of storage, and information
that is kept only temporarily to perform a communications function is
not subject to regulation (for example, active queue management, deep
packet inspection). Such Measurement Tasks could reveal all the
websites a Subscriber visits and the applications and/or services
they use. This issue is not specific to LMAP. For instance, IPFIX
has discussed similar issues (see section 11.8 of [RFC7011]), but
mitigations described in the sections below were considered beyond
their scope.

Other types of Measurement Task are conducted on traffic which is
created specifically for the purpose. Even if a user host generates
Measurement Traffic, there is limited sensitive information about the
Subscriber present and stored in the Measurement System:

o IP address in use (and possibly sub-IP addresses and names)

o Status as a study volunteer and Schedule of Measurement Tasks

On the other hand, for a service provider the sensitive information
like Measurement Results is the same for all Measurement Tasks.

From the Subscriber perspective, both types of Measurement Task
potentially expose the description of Internet access service and
specific service parameters, such as subscribed rate and type of
access.

8.4. Privacy analysis of the communication models

   This section examines each of the protocol exchanges described at a
   high level in Section 5 and some example Measurement Tasks, and
   identifies specific sensitive information which must be secured
   during communication for each case. With the protocol-related
   sensitive information identified, we can better consider the threats
   described in the following section.

   From the privacy perspective, all entities participating in LMAP
   protocols can be considered "observers" according to the definition
   in [RFC6973]. Their stored information potentially poses a threat to
   privacy, especially if one or more of these functional entities has
   been compromised. Likewise, all devices on the paths used for
   control, reporting, and measurement are also observers.

8.4.1. MA Bootstrapping

   Section 5.1 provides the communication model for the Bootstrapping
   process.

Although the specification of mechanisms for Bootstrapping the MA are beyond the initial LMAP work scope, designers should recognize that the Bootstrapping process is extremely powerful and could cause an MA to join a new or different LMAP system with a different Controller and Collector, or simply install new Metrics with associated Measurement Methods (for example to record DNS queries). A Bootstrap attack could result in a breach of the LMAP system with significant sensitive information exposure depending on the capabilities of the MA, so sufficient security protections are warranted.

The Bootstrapping process provides sensitive information about the LMAP system and the organisation that operates it, such as

o the MA's identifier (MA-ID)

o the address that identifies the Control Channel, such as the Controller's FQDN

o Security information for the Control Channel

During the Bootstrap process for an MA located at a single subscriber's service demarcation point, the MA receives a MA-ID which is a persistent pseudonym for the Subscriber. Thus, the MA-ID is considered sensitive information because it could provide the link between Subscriber identification and Measurements Results.

Also, the Bootstrap process could assign a Group-ID to the MA. The specific definition of information represented in a Group-ID is to be determined, but several examples are envisaged including use as a pseudonym for a set of Subscribers, a class of service, an access technology, or other important categories. Assignment of a Group-ID enables anonymisation sets to be formed on the basis of service type/grade/rates. Thus, the mapping between Group-ID and MA-ID is considered sensitive information.

8.4.2. Controller <-> Measurement Agent

The high-level communication model for interactions between the LMAP Controller and Measurement Agent is illustrated in Section 5.2. The primary purpose of this exchange is to authenticate and task a Measurement Agent with Measurement Instructions, which the Measurement Agent then acts on autonomously.

Primarily IP addresses and pseudonyms (MA-ID, Group-ID) are exchanged with a capability request, then measurement-related information of interest such as the parameters, schedule, metrics, and IP addresses of measurement devices. Thus, the measurement Instruction contains sensitive information which must be secured. For example, the fact

that an ISP is running additional measurements beyond the set
reported externally is sensitive information, as are the additional
Measurements Tasks themselves. The Measurement Schedule is also
sensitive, because an attacker intending to bias the results without
being detected can use this information to great advantage.

An organisation operating the Controller having no service
relationship with a user who hosts the Measurement Agent *could* gain
real-name mapping to a public IP address through user participation
in an LMAP system (this applies to the Measurement Collection
protocol, as well).

8.4.3. Collector <-> Measurement Agent

The high-level communication model for interactions between the
Measurement Agent and Collector is illustrated in Section 5.4. The
primary purpose of this exchange is to authenticate and collect
Measurement Results from a MA, which the MA has measured autonomously
and stored.

The Measurement Results are the additional sensitive information
included in the Collector-MA exchange. Organisations collecting LMAP
measurements have the responsibility for data control. Thus, the
Results and other information communicated in the Collector protocol
must be secured.

8.4.4. Measurement Peer <-> Measurement Agent

A Measurement Method involving Measurement Traffic raises potential
privacy issues, although the specification of the mechanisms is
beyond the scope of the initial LMAP work. The high-level
communications model below illustrates the various exchanges to
execute such a Measurement Method and store the Results.

We note the potential for additional observers in the figures below
by indicating the possible presence of a NAT, which has additional
significance to the protocols and direction of initiation.

The various messages are optional, depending on the nature of the
Measurement Method. It may involve sending Measurement Traffic from
the Measurement Peer to MA, MA to Measurement Peer, or both.
Similarly, a second (or more) MAs may be involved. (Note: For
simplicity, the Figure and description don't show the non-LMAP
functionality that is associated with the transfer of the Measurement
Traffic and is located at the devices with the MA and MP.)

```
 _____                        _____
|                   |                      |                   |
|Measurement Peer   |========== NAT ? =========|Measurement Agent|
|_____|                      |_____|

                               <-                 (Key Negotiation &
                                                   Encryption Setup)
(Encrypted Channel             ->
Established)
(Announce capabilities         ->
& status)
                               <-                 (Select capabilities)
ACK                            ->
                               <-                 (Measurement Request
                                                   (MA+MP IPAddrs,set of
                                                    Metrics, Schedule))

ACK                            ->

Measurement Traffic            <>                 Measurement Traffic
(may/may not be encrypted)              (may/may not be encrypted)

                               <-                 (Stop Measurement Task)

Measurement Results            ->
(if applicable)
                               <-                 ACK, Close
```

This exchange primarily exposes the IP addresses of measurement
devices and the inference of measurement participation from such
traffic. There may be sensitive information on key points in a
service provider's network included. There may also be access to
measurement-related information of interest such as the Metrics,
Schedule, and intermediate results carried in the Measurement Traffic
(usually a set of timestamps).

The Measurement Peer may be able to use traffic analysis (perhaps
combined with traffic injection) to obtain interesting insights about
the Subscriber. As a simple example, if the Measurement Task
includes a pre-check that the end-user isn't already sending traffic,
the Measurement Peer may be able to deduce when the Subscriber is
away on holiday, for example.

If the Measurement Traffic is unencrypted, as found in many systems
today, then both timing and limited results are open to on-path
observers.

8.4.5. Measurement Agent

   Some Measurement Methods only involve a single Measurement Agent
   observing existing traffic. They raise potential privacy issues,
   although the specification of the mechanisms is beyond the scope of
   the initial LMAP work.

   The high-level communications model below illustrates the collection
   of user information of interest with the Measurement Agent performing
   the monitoring and storage of the Results. This particular exchange
   is for measurement of DNS Response Time, which most frequently uses
   UDP transport. (Note: For simplicity, the Figure and description
   don't show the non-LMAP functionality that is associated with the
   transfer of the Measurement Traffic and is located at the devices
   with the MA.)

```
 _____                            _____
|                  |                          |          |
|   DNS Server     |========== NAT ? =========*======| User client|
|_____|                          ^       |_____|
                                        _____|_____
                                       |               |
                                       | Measurement   |
                                       |    Agent       |
                                       |_____|


                        <-                Name Resolution Req
                                          (MA+MP IPAddrs,
                                          Desired Domain Name)
Return Record                  ->
```
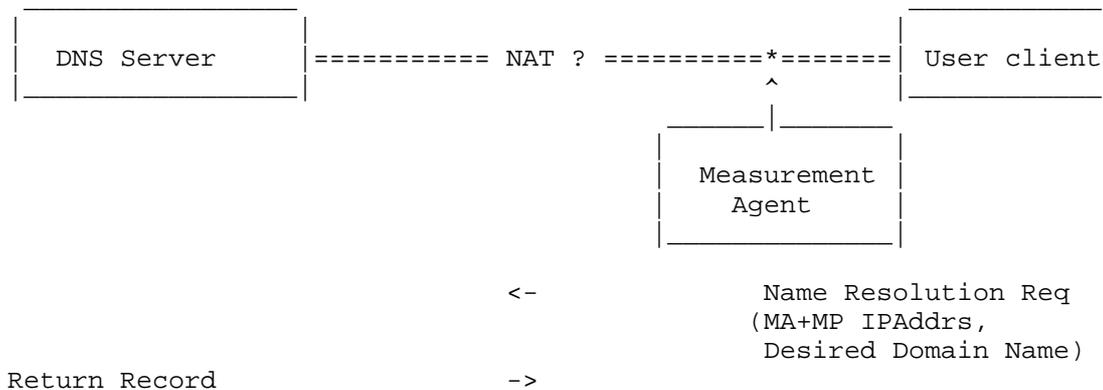
   In this particular example, the MA monitors DNS messages in order to
   measure that DNS response time. The Measurement Agent may be
   embedded in the user host, or it may be located in another device
   capable of observing user traffic. The MA learns the IP addresses of
   measurement devices and the intent to communicate with or access the
   services of a particular domain name, and perhaps also information on
   key points in a service provider's network, such as the address of
   one of its DNS servers.

   In principle, any of the user sensitive information of interest
   (listed above) can be collected and stored in the monitoring scenario
   and so must be secured.

   It would also be possible for a Measurement Agent to source the DNS
   query itself. But then there are few privacy concerns.

8.4.6. Storage and reporting of Measurement Results

   Although the mechanisms for communicating results (beyond the initial
   Collector) are beyond the initial LMAP work scope, there are
   potential privacy issues related to a single organisation's storage
   and reporting of Measurement Results. Both storage and reporting
   functions can help to preserve privacy by implementing the
   mitigations described below.

8.5. Threats

   This section indicates how each of the threats described in [RFC6973]
   apply to the LMAP entities and their communication and storage of
   "information of interest". Denial of Service (DOS) and other attacks
   described in the Security section represent threats as well, and
   these attacks are more effective when sensitive information
   protections have been compromised.

8.5.1. Surveillance

   Section 5.1.1 of [RFC6973] describes Surveillance as the "observation
   or monitoring of and individual's communications or activities."
   Hence all Measurement Methods that measure user traffic are a form of
   surveillance, with inherent risks.

   Measurement Methods which avoid periods of user transmission
   indirectly produce a record of times when a subscriber or authorised
   user has used their network access service.

   Measurement Methods may also utilise and store a Subscriber's
   currently assigned IP address when conducting measurements that are
   relevant to a specific Subscriber. Since the Measurement Results are
   time-stamped, they could provide a record of IP address assignments
   over time.

   Either of the above pieces of information could be useful in
   correlation and identification, described below.

8.5.2. Stored data compromise

   Section 5.1.2 of [RFC6973] describes Stored Data Compromise as
   resulting from inadequate measures to secure stored data from
   unauthorised or inappropriate access. For LMAP systems this includes
   deleting or modifying collected measurement records, as well as data
   theft.

   The primary LMAP entity subject to compromise is the repository,
   which stores the Measurement Results; extensive security and privacy

threat mitigations are warranted. The Collector and MA also store
sensitive information temporarily, and need protection. The
communications between the local storage of the Collector and the
repository is beyond the scope of the initial LMAP work, though this
communications channel will certainly need protection as well as the
mass storage itself.

The LMAP Controller may have direct access to storage of Subscriber
information (location, billing, service parameters, etc.) and other
information which the controlling organisation considers private, and
again needs protection.

Note that there is tension between the desire to store all raw
results in the LMAP Collector (for reproducibility and custom
analysis), and the need to protect the privacy of measurement
participants. Many of the compromise mitigations described in
section 8.6 below are most efficient when deployed at the MA,
therefore minimising the risks with stored results.

8.5.3. Correlation and identification

Sections 5.2.1 and 5.2.2 of [RFC6973] describe Correlation as
combining various pieces of information to obtain desired
characteristics of an individual, and Identification as using this
combination to infer identity.

The main risk is that the LMAP system could unwittingly provide a key
piece of the correlation chain, starting with an unknown Subscriber's
IP address and another piece of information. For example, a
Subscriber utilised Internet access from 2000 to 2310 UTC, because
the Measurement Tasks were deferred, or sent a name resolution for
www.example.com at 2300 UTC.

If a user's access with another system already gave away sensitive
info, correlation is clearly easier and can result in re-
identification, even when an LMAP conserves sensitive information to
great extent.

8.5.4. Secondary use and disclosure

Sections 5.2.3 and 5.2.4 of [RFC6973] describes Secondary Use as
unauthorised utilisation of an individual's information for a purpose
the individual did not intend, and Disclosure is when such
information is revealed causing other's notions of the individual to
change, or confidentiality to be violated.

Measurement Methods that measure user traffic are a form of Secondary
Use, and the Subscribers' permission should be obtained beforehand.

It may be necessary to obtain the measured ISP's permission to
conduct measurements, for example when required by the terms and
conditions of the service agreement, and notification is considered
good measurement practice.

For Measurement Methods that measure Measurement Traffic the
Measurement Results provide some limited information about the
Subscriber or ISP and could result in Secondary Uses. For example,
the use of the Results in unauthorised marketing campaigns would
qualify as Secondary Use. Secondary use may break national laws and
regulations, and may violate individual's expectations or desires.

8.6. Mitigations

   This section examines the mitigations listed in section 6 of
   [RFC6973] and their applicability to LMAP systems. Note that each
   section in [RFC6973] identifies the threat categories that each
   technique mitigates.

8.6.1. Data minimisation

   Section 6.1 of [RFC6973] encourages collecting and storing the
   minimal information needed to perform a task.

   LMAP results can be useful for general reporting about performance
   and for specific troubleshooting. They need different levels of
   information detail, as explained in the paragraphs below.

   For general results, the results can be aggregated into large
   categories (the month of March, all subscribers West of the
   Mississippi River). In this case, all individual identifications
   (including IP address of the MA) can be excluded, and only relevant
   results are provided. However, this implies a filtering process to
   reduce the information fields, because greater detail was needed to
   conduct the Measurement Tasks in the first place.

   For troubleshooting, so that a network operator or end user can
   identify a performance issue or failure, potentially all the network
   information (IP addresses, equipment IDs, location), Measurement
   Schedule, service configuration, Measurement Results, and other
   information may assist in the process. This includes the information
   needed to conduct the Measurements Tasks, and represents a need where
   the maximum relevant information is desirable, therefore the greatest
   protections should be applied. This level of detail is greater than
   needed for general performance monitoring.

   As regards Measurement Methods that measure user traffic, we note
   that a user may give temporary permission (to enable detailed

troubleshooting), but withhold permission for them in general. Here
the greatest breadth of sensitive information is potentially exposed,
and the maximum privacy protection must be provided. The Collector
may perform pre-storage minimisation and other mitigations (below) to
help preserve privacy.

For MAs with access to the sensitive information of users (e.g.,
within a home or a personal host/handset), it is desirable for the
results collection to minimise the data reported, but also to balance
this desire with the needs of troubleshooting when a service
subscription exists between the user and organisation operating the
measurements.

8.6.2. Anonymity

Section 6.1.1 of [RFC6973] describes a way in which anonymity is
achieved: "there must exist a set of individuals that appear to have
the same attributes as the individual", defined as an "anonymity
set".

Experimental methods for anonymisation of user identifiable data (and
so particularly applicable to Measurement Methods that measure user
traffic) have been identified in [RFC6235]. However, the findings of
several of the same authors is that "there is increasing evidence
that anonymisation applied to network trace or flow data on its own
is insufficient for many data protection applications as in [Bur10]."
Essentially, the details of such Measurement Methods can only be
accessed by closed organisations, and unknown injection attacks are
always less expensive than the protections from them. However, some
forms of summary may protect the user's sensitive information
sufficiently well, and so each Metric must be evaluated in the light
of privacy.

The techniques in [RFC6235] could be applied more successfully in
Measurement Methods that generate Measurement Traffic, where there
are protections from injection attack. The successful attack would
require breaking the integrity protection of the LMAP Reporting
Protocol and injecting Measurement Results (known fingerprint, see
section 3.2 of [RFC6973]) for inclusion with the shared and
anonymised results, then fingerprinting those records to ascertain
the anonymisation process.

Beside anonymisation of measured Results for a specific user or
provider, the value of sensitive information can be further diluted
by summarising the results over many individuals or areas served by
the provider. There is an opportunity enabled by forming anonymity
sets [RFC6973] based on the reference path measurement points in
[RFC7398]. For example, all measurements from the Subscriber device

can be identified as "mp000", instead of using the IP address or
other device information. The same anonymisation applies to the
Internet Service Provider, where their Internet gateway would be
referred to as "mp190".

Another anonymisation technique is for the MA to include its Group-ID
instead of its MA-ID in its Measurement Reports, with several MAs
sharing the same Group-ID.

8.6.3. Pseudonymity

Section 6.1.2 of [RFC6973] indicates that pseudonyms, or nicknames,
are a possible mitigation to revealing one's true identity, since
there is no requirement to use real names in almost all protocols.

A pseudonym for a measurement device's IP address could be an LMAP-
unique equipment ID. However, this would likely be a permanent
handle for the device, and long-term use weakens a pseudonym's power
to obscure identity.

8.6.4. Other mitigations

Data can be de-personalised by blurring it, for example by adding
synthetic data, data-swapping, or perturbing the values in ways that
can be reversed or corrected.

Sections 6.2 and 6.3 of [RFC6973] describe User Participation and
Security, respectively.

Where LMAP measurements involve devices on the Subscriber's premises
or Subscriber-owned equipment, it is essential to secure the
Subscriber's permission with regard to the specific information that
will be collected. The informed consent of the Subscriber (and, if
different, the end user) may be needed, including the specific
purpose of the measurements. The approval process could involve
showing the Subscriber their measured information and results before
instituting periodic collection, or before all instances of
collection, with the option to cancel collection temporarily or
permanently.

It should also be clear who is legally responsible for data
protection (privacy); in some jurisdictions this role is called the
'data controller'. It is always good practice to limit the time of
personal information storage.

Although the details of verification would be impenetrable to most
subscribers, the MA could be architected as an "app" with open
source-code, pre-download and embedded terms of use and agreement on

measurements, and protection from code modifications usually provided by the app-stores. Further, the app itself could provide data reduction and temporary storage mitigations as appropriate and certified through code review.

LMAP protocols, devices, and the information they store clearly need to be secure from unauthorised access. This is the hand-off between privacy and security considerations (Section 7). The Data Controller has the (legal) responsibility to maintain data protections described in the Subscriber's agreement and agreements with other organisations.

Finally, it is recommended that each entity in section 8.1, (individuals, ISPs, Regulators, others) assess the risks of LMAP data collection by conducting audits of their data protection methods.

9. IANA considerations

There are no IANA considerations in this memo.

10. Acknowledgments

This document originated as a merger of three individual drafts: draft-eardley-lmap-terminology-02, draft-akhter-lmap-framework-00, and draft-eardley-lmap-framework-02.

Thanks to Juergen Schoenwaelder for his detailed review of the terminology. Thanks to Charles Cook for a very detailed review of -02. Thanks to Barbara Stark and Ken Ko for many helpful comments about later versions.

Thanks to numerous people for much discussion, directly and on the LMAP list (apologies to those unintentionally omitted): Alan Clark, Alissa Cooper, Andrea Soppera, Barbara Stark, Benoit Claise, Brian Trammell, Charles Cook, Dan Romascanu, Dave Thorne, Frode Soerensen, Greg Mirsky, Guangqing Deng, Jason Weil, Jean-Francois Tremblay, Jerome Benoit, Joachim Fabini, Juergen Schoenwaelder, Jukka Manner, Ken Ko, Lingli Deng, Mach Chen, Matt Mathis, Marc Ibrahim, Michael Bugenhagen, Michael Faath, Nalini Elkins, Radia Perlman, Rolf Winter, Sam Crawford, Sharam Hakimi, Steve Miller, Ted Lemon, Timothy Carey, Vaibhav Bajpai, Vero Zheng, William Lupton.

Philip Eardley, Trevor Burbridge and Marcelo Bagnulo 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.

11. History

   First WG version, copy of draft-folks-lmap-framework-00.

11.1. From -00 to -01

   o new sub-section of possible use of Group-IDs for privacy

   o tweak to definition of Control protocol

   o fix typo in figure in S5.4

11.2. From -01 to -02

   o change to INFORMATIONAL track (previous version had typo'd
      Standards track)

   o new definitions for Capabilities Information and Failure
      Information

   o clarify that diagrams show LMAP-level information flows.
      Underlying protocol could do other interactions, eg to get through
      NAT or for Collector to pull a Report

   o add hint that after a re-boot should pause random time before re-
      register (to avoid mass calling event)

   o delete the open issue "what happens if a Controller fails" (normal
      methods can handle)

   o add some extra words about multiple Tasks in one Schedule

   o clarify that new Schedule replaces (rather than adds to) and old
      one. Similarly for new configuration of Measurement Tasks or
      Report Channels.

   o clarify suppression is temporary stop; send a new Schedule to
      permanently stop Tasks

   o alter suppression so it is ACKed

   o add un-suppress message

   o expand the text on error reporting, to mention Reporting failures
      (as well as failures to action or execute Measurement Task &
      Schedule)

   o add some text about how to have Tasks running indefinitely

o add that optionally a Report is not sent when there are no
   Measurement Results

o add that a Measurement Task may create more than one Measurement
   Result

o clarify /amend /expand that Reports include the "raw" Measurement
   Results - any pre-processing is left for lmap2.0

o add some cautionary words about what if the Collector unexpectedly
   doesn't hear from a MA

o add some extra words about the potential impact of Measurement
   Tasks

o clarified various aspects of the privacy section

o updated references

o minor tweaks

11.3. From -02 to -03

o alignment with the Information Model [burbridge-lmap-information-
   model] as this is agreed as a WG document

o One-off and periodic Measurement Schedules are kept separate, so
   that they can be updated independently

o Measurement Suppression in a separate sub-section. Can now
   optionally include particular Measurement Tasks &/or Schedules to
   suppress, and start/stop time

o for clarity, concept of Channel split into Control, Report and MA-
   to-Controller Channels

o numerous editorial changes, mainly arising from a very detailed
   review by Charles Cook

o

11.4. From -03 to -04

o updates following the WG Last Call, with the proposed consensus on
   the various issues as detailed in
   http://tools.ietf.org/agenda/89/slides/slides-89-lmap-2.pdf. In
   particular:

o tweaked definitions, especially of Measurement Agent and
   Measurement Peer

o Instruction - left to each implementation & deployment of LMAP to
   decide on the granularity at which an Instruction Message works

o words added about overlapping Measurement Tasks (Measurement
   System can handle any way they choose; Report should mention if
   the Task overlapped with another)

o Suppression: no defined impact on Passive Measurement Task; extra
   option to suppress on-going Active Measurement Tasks; suppression
   doesn't go to Measurement Peer, since they don't understand
   Instructions

o new concept of Data Transfer Task (and therefore adjustment of the
   Channel concept)

o enhancement of Results with Subscriber's service parameters -
   could be useful, don't define how but can be included in Report to
   various other sections

o various other smaller improvements, arising from the WGLC

o Appendix added with examples of Measurement Agents and Peers in
   various deployment scenarios. To help clarify what these terms
   mean.

11.5. From -04 to -05

o clarified various scoping comments by using the phrase "scope of
   initial LMAP work" (avoiding "scope of LMAP WG" since this may
   change in the future)

o added a Configuration Protocol - allows the Controller to update
   the MA about information that it obtained during the bootstrapping
   process (for consistency with Information Model)

o Removed over-detailed information about the relationship between
   the different items in Instruction, as this seems more appropriate
   for the information model. Clarified that the lists given are
   about the aims and not a list of information elements (these will
   be defined in draft-ietf-information-model).

o the Measurement Method, specified as a URI to a registry entry -
   rather than a URN

   o MA configured with time limit after which, if it hasn't heard from
      Controller, then it stops running Measurement Tasks (rather than
      this being part of a Schedule)

   o clarified there is no distinction between how capabilities,
      failure and logging information are transferred (all can be when
      requested by Controller or by MA on its own initiative).

   o removed mention of Data Transfer Tasks. This abstraction is left
      to the information model i-d

   o added Deployment sub-section about Measurement Agent embedded in
      ISP Network

   o various other smaller improvements, arising from the 2nd WGLC

11.6. From -05 to -06

   o clarified terminlogy around Measurement Methods and Tasks. Since
      within a Method there may be several different roles (requester
      and responder, for instance)

   o Suppression: there is now the concept of a flag (boolean) which
      indicates whether a Task is by default gets suppressed or not.
      The optional suppression message (with list of specific tasks
      /schedules to suppress) over-rides this flag.

   o The previous bullet also means there is no need to make a
      distinction between active and passive Measurement Tasks, so this
      distinction is removed.

   o removed Configuration Protocol - Configuration is part of the
      Instruction and so uses the Control Protocol.

11.7. From -06 to -07

   o Clarifications and nits

11.8. From -07 to -08

   o Clarifications resulting from WG 3rd LC, as discussed in
      https://tools.ietf.org/agenda/90/slides/slides-90-lmap-0.pdf, plus
      comments made in the IETF-90 meeting.

   o added mention of "measurement point designations" in Measurement
      Task configuration and Report Protocol.

11.9. From -08 to -09

   o Clarifications and changes from the AD review (Benoit Claise) and
      security directorate review (Radia Perlman).

11.10. From -09 to -10

   o More changes from the AD review (Benoit Claise).

11.11. From -10 to -11

   o More changes from the AD review (Benoit Claise).

11.12. From -11 to -12

   o Fixing nits from IETF Last call and authors.

11.13. From -12 to -13

   o IESG changes.

11.14. From -13 to -14

   o Fixing Figure 1.

12. Informative References

   [Bur10]   Burkhart, M., Schatzmann, D., Trammell, B., and E. Boschi,
              "The Role of Network Trace anonymisation Under Attack",
              January 2010.

   [TR-069]   TR-069, , "CPE WAN Management Protocol",
              http://www.broadband-forum.org/technical/trlist.php,
              November 2013.

   [UPnP]   ISO/IEC 29341-x, , "UPnP Device Architecture and UPnP
              Device Control Protocols specifications",
              http://upnp.org/sdcps-and-certification/standards/, 2011.

   [RFC1035] Mockapetris, P., "Domain names - implementation and
              specification", STD 13, RFC 1035, November 1987.

   [RFC4101] Rescorla, E. and IAB, "Writing Protocol Models", RFC 4101,
              June 2005.

   [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally
              Unique IDentifier (UUID) URN Namespace", RFC 4122, July
              2005.

   [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A.
              Bierman, "Network Configuration Protocol (NETCONF)", RFC
              6241, June 2011.

   [RFC7011] Claise, B., Trammell, B., and P. Aitken, "Specification of
              the IP Flow Information Export (IPFIX) Protocol for the
              Exchange of Flow Information", STD 77, RFC 7011, September
              2013.

   [RFC7368] Chown, T., Arkko, J., Brandt, A., Troan, O., and J. Weil,
              "IPv6 Home Networking Architecture Principles", RFC 7368,
              October 2014.

   [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an
              Attack", BCP 188, RFC 7258, May 2014.

   [I-D.ietf-lmap-use-cases]
              Linsner, M., Eardley, P., Burbridge, T., and F. Sorensen,
              "Large-Scale Broadband Measurement Use Cases", draft-ietf-
              lmap-use-cases-06 (work in progress), February 2015.

   [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.

   [RFC6419] Wasserman, M. and P. Seite, "Current Practices for
              Multiple-Interface Hosts", RFC 6419, November 2011.

   [RFC6887] Wing, D., Cheshire, S., Boucadair, M., Penno, R., and P.
              Selkirk, "Port Control Protocol (PCP)", RFC 6887, April
              2013.

   [I-D.ietf-lmap-information-model]
              Burbridge, T., Eardley, P., Bagnulo, M., and J.
              Schoenwaelder, "Information Model for Large-Scale
              Measurement Platforms (LMAP)", draft-ietf-lmap-
              information-model-05 (work in progress), April 2015.

   [RFC6235] Boschi, E. and B. Trammell, "IP Flow Anonymization
              Support", RFC 6235, May 2011.

   [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J.,
              Morris, J., Hansen, M., and R. Smith, "Privacy
              Considerations for Internet Protocols", RFC 6973, July
              2013.

   [RFC4656] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and M.
              Zekauskas, "A One-way Active Measurement Protocol
              (OWAMP)", RFC 4656, September 2006.

   [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J.
              Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)",
              RFC 5357, October 2008.

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

   [RFC7398] Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and
              A. Morton, "A Reference Path and Measurement Points for
              Large-Scale Measurement of Broadband Performance", RFC
              7398, February 2015.

Authors' Addresses

   Philip Eardley
   BT
   Adastral Park, Martlesham Heath
   Ipswich
   ENGLAND

   Email: philip.eardley@bt.com

   Al Morton
   AT&T Labs
   200 Laurel Avenue South
   Middletown, NJ
   USA

   Email: acmorton@att.com

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

   Phone: 34 91 6249500
   Email: marcelo@it.uc3m.es
   URI: http://www.it.uc3m.es

Trevor Burbridge
BT
Adastral Park, Martlesham Heath
Ipswich
ENGLAND

Email: trevor.burbridge@bt.com

Paul Aitken
Brocade
Edinburgh, Scotland
UK

Email: paitken@brocade.com

Aamer Akhter
Consultant
118 Timber Hitch
Cary, NC
USA

Email: aakhter@gmail.com

Network Working Group   T. Burbridge
Internet-Draft   P. Eardley
Intended status: Standards Track   BT
Expires: October 23, 2017   M. Bagnulo
                                    Universidad Carlos III de Madrid
                                               J. Schoenwaelder
                                             Jacobs University Bremen
                                                     April 21, 2017

Information Model for Large-Scale Measurement Platforms (LMAP)
                  draft-ietf-lmap-information-model-18

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 Measurement Agent 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 Measurement
   Agent 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 23, 2017.

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. A typical use case is the execution of
   broadband measurements [RFC7536]. 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-scale measurement
platform as used in this document is described in detail in
[RFC7594].

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 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). A YANG data model implementing the information model can be
found in [I-D.ietf-lmap-yang].

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

2. Notation

   This document uses a programming language-like notation to define the
   properties of the objects of the information model. An optional
   property is enclosed by square brackets, [ ], and a list property 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.

   The object definitions use a couple of base types that are defined as
   follows:

   int   A type representing signed or unsigned integer numbers.
               This information model does not define a precision nor
               does it make a distinction between signed and unsigned
               number ranges. This type is also used to represent
               enumerations.

   boolean   A type representing a boolean value.

   string   A type representing a human-readable string consisting of
               a (possibly restricted) subset of Unicode and ISO/IEC
               10646 [ISO.10646] characters.

   datetime   A type representing a date and time using the Gregorian
               calendar. The datetime format MUST conform to RFC 3339
               [RFC3339].

   uuid   A type representing Universally Unique IDentifier (UUID(
               as defined in RFC 4122 [RFC4122]. The UUID values are
               expected to be unique within an installation of a large-
               scale measurement system.

   uri   A type representing a Uniform Resource Identifier as
               defined in STD 66 [RFC3986].

   ip-address A type representing an IP address. This type supports
               both IPv4 and IPv6 addresses.

   counter   A non-negative integer that monotonically increases.
               Counters may have discontinuities and they are not
               expected to persist across restarts.

   credentials An opaque type representing credentials needed by a
               cryptographic mechanism to secure communication. Data

        models must expand this opaque type as needed and
        required by the security protocols utilized.

    data   An opaque type representing data obtained from
               measurements.

    Names of objects are generally assumed to be unique within an
    implementation.

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 [RFC7594]. 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.

    The information model is divided into six aspects. 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.

    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 Measurement
        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 Measurement 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 Event and 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 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 modeled by the six aspects of the information model
is supported by a number of common information objects. These
objects are also described later in this document and comprise of:

a. Schedules. A set of Schedules tells the MA to execute Actions.
    An Action of a Schedule leads to the execution of a Task.
    Without a Schedule no Task (including measurements or 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. A specific Schedule
can only be active once. Attempts to start a Schedule while the
same Schedule is still running will fail.

b. 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.

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

d. Events. A set of Event objects that can be referenced from the
    Schedules. Each Schedule always references exactly one Event
    object that determines when the schedule is executed. An Event
    object specifies either a singleton or series of events that
    indicate when Tasks should be executed. A commonly used kind of
    Event objects are Timing objects. For Event objects specifying a
    series of events, it is generally a good idea to configure an end
    time and to refresh the end time as needed to ensure that MAs
    that loose connectivity to their controller do not continue
    executing Schedules forever.

Figure 1 illustrates the structure in which these common information
objects are referenced. The references are achieved by each object
(Task Configuration, Event) being given a short textual 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
       |-- triggered by --> Event
       |
       |-- executes --> Action 1
       |                    |-- using --> Task Configuration
       |                    |
       |                    '-- feeding to --> Destination Schedule
       :
       :
       '-- executes --> Action N
                            |-- using --> Task Configuration
                            |
                            '-- feeding to --> Destination Schedule
```
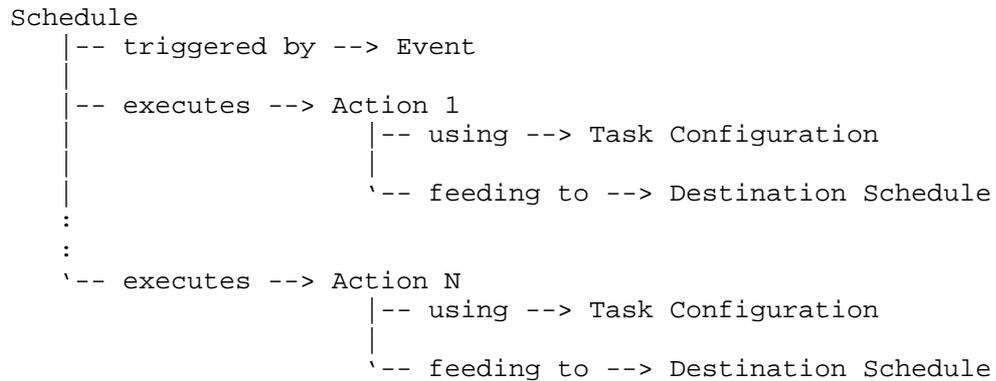
         Figure 1: Relationship between Schedules, Events, Actions, Task
                 Configurations, and Destination Schedules

   The primary function of an MA is to execute Schedules. A Schedule,
   which is triggered by an Event, executes a number of Actions. An
   Action refers to a Configured Task and it may feed results to a
   Destination Schedule. Both, Actions and Configured Tasks can provide
   parameters, represented as Action Options and Task Options.

   Tasks can implement a variety of different functions. 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 support the communication with a Controller
       and Collectors:

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

       B. Control Task(s) implement the Control Protocol and
           communicate with the Controller.

   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

Figure 1 indicates that Actions can produce data that is fed into
Destination Schedules. This can by used by Actions implementing
Measurement Tasks to feed measurement results to a Schedule that
triggers Actions implementing Reporting Tasks. Data fed to a
Destination Schedule is consumed by the first Action of the
Destination Schedule if the Destination Schedule is using sequential
or pipelined execution mode and it is consumed by all Actions of the
Destination Schedule if the Destination Schedule is using parallel
execution mode.

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 represented as an Event that invokes 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, as
defined by a Task Configuration.

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-preconfig-agent-id;]
     ma-task-obj         ma-preconfig-control-tasks<1..*>;
     ma-channel-obj      ma-preconfig-control-channels<1..*>;
     ma-schedule-obj     ma-preconfig-control-schedules<1..*>;
    [uri                 ma-preconfig-device-id;]
     credentials         ma-preconfig-credentials;
} ma-preconfig-obj;
```

The ma-preconfig-obj describes information that needs to be available
to the MA in order to bootstrap communication with a Controller. The
ma-preconfig-obj consists of the following elements:

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

ma-preconfig-control-tasks: An unordered set of task objects.

ma-preconfig-control-channels: An unordered set of channel objects.

ma-preconfig-control-schedules: An unordered set of scheduling
                               objects.

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

ma-preconfig-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. Additional flags control whether the
MA ID or the Group ID are included in Reports. The reporting of a
Group ID without the MA ID may allow 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 seconds
since the last communication with the Controller after which an Event
is generated that can trigger the suspension of Instruction
Schedules. 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.

3.2.1. Definition of ma-config-obj

```
object {
    uuid               ma-config-agent-id;
    ma-task-obj         ma-config-control-tasks<1..*>;
    ma-channel-obj      ma-config-control-channels<1..*>;
    ma-schedule-obj     ma-config-control-schedules<1..*>;
    credentials         ma-config-credentials;
   [string              ma-config-group-id;]
   [string              ma-config-measurement-point;]
   [boolean             ma-config-report-agent-id;]
   [boolean             ma-config-report-group-id;]
   [boolean             ma-config-report-measurement-point;]
   [int                 ma-config-controller-timeout;]
} ma-config-obj;
```

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

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

ma-config-control-tasks: An unordered set of task objects.

ma-config-control-channels: An unordered set of channel
                                    objects.

ma-config-control-schedules: An unordered set of scheduling
                                    objects.

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

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

ma-config-measurement-point: An optional identifier for the
                                    measurement point indicating
                                    where the measurement agent is
                                    located on a path (see [RFC7398]
                                    for further details).

ma-config-report-agent-id: An optional flag indicating
                                    whether the agent identifier (ma-
                                    config-agent-id) is included in
                                    reports. The default value is
                                    true.

ma-config-report-group-id: An optional flag indicating
                                    whether the group identifier (ma-
                                    config-group-id) is included in
                                    reports. The default value is
                                    false.

ma-config-report-measurement-point: An optional flag indicating
                                    whether the measurement point
                                    (ma-config-measurement-point)
                                    should be included in reports.
                                    The default value is false.

ma-config-controller-timeout: A timer is started after each
                                    successful contact with a
                                    controller. When the timer
                                    reaches the controller-timeout
                                    (measured in seconds), an event
                                    is raised indicating that
                                    connectivity to the controller
                                    has been lost (see ma-controller-
                                    lost-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 the
   Actions of Instruction Schedules that specify when to execute them.
   The results can be communicated to other Schedules 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 a 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 Actions to execute according to a
given triggering Event. An Action extends a Configured Task with
additional specific parameters. An Event can trigger the execution
of a single Action or it can trigger a repeated series of Actions.
The Schedule also specifies how to link Tasks output data to other
Schedules.

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.

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).

A single Suppression object is able to enable/disable a set of
Instruction Tasks that are tagged for suppression. This enables fine
grained control on which Tasks are suppressed. Suppression of both
matching Actions and Measurement Schedules is supported. Support for
disabling specific Actions allows malfunctioning or mis-configured
Tasks or Actions 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.

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 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-instruction-channels<0..*>;
    ma-schedule-obj    ma-instruction-schedules<0..*>;
   [ma-suppression-obj  ma-instruction-suppressions<0..*>;]
} ma-instruction-obj;
```

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

ma-instruction-tasks: A possibly empty unordered set of task
                             objects.

      ma-instruction-channels: A possibly empty unordered set of
                                     channel objects.

      ma-instruction-schedules: A possibly empty unordered set of
                                     schedule objects.

      ma-instruction-suppressions: An optional possibly empty unordered
                                     set of suppression objects.

3.3.2. Definition of ma-suppression-obj

      object {
           string   ma-suppression-name;
          [ma-event-obj   ma-suppression-start;]
          [ma-event-obj   ma-suppression-end;]
          [string   ma-suppression-match<0..*>;]
          [boolean   ma-suppression-stop-running;]
      } ma-suppression-obj;

   The ma-suppression-obj controls the suppression of schedules or
   actions and consists of the following elements:

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

      ma-suppression-start: The optional event indicating when
                                     suppression starts. If not present,
                                     the suppression starts immediately,
                                     i.e., as if the value would be
                                     'immediate'.

      ma-suppression-end: The optional event indicating when
                                     suppression ends. If not present, the
                                     suppression does not have a defined
                                     end, i.e., the suppression remains for
                                     an indefinite period of time.

      ma-suppression-match: An optional and possibly empty
                                     unordered set of match patterns. The
                                     suppression will apply to all schedules
                                     (and their actions) that have a
                                     matching value in their ma-schedule-
                                     suppression-tags and all actions that
                                     have a matching value in their ma-
                                     action-suppression-tags. Pattern
                                     matching is done using glob style
                                     pattern (see below).

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

Glob style pattern matching is following POSIX.2 fnmatch() [POSIX.2]
without special treatment of file paths:

              *   matches a sequence of characters
              ?   matches a single character
              [seq]   matches any character in seq
              [!seq]   matches any character not in seq

A backslash followed by a character matches the following character.
In particular:

              *   matches *
              \?   matches ?
              \\   matches \

A sequence seq may be a sequence of characters (e.g., [abc] or a
range of characters (e.g., [a-c]).

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;
    int               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 registry entries) 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-capability-obj

```
object {
    string                ma-capability-hardware;
    string                ma-capability-firmware;
    string                ma-capability-version;
    [string               ma-capability-tags<0..*>;]
    [ma-capability-task-obj  ma-capability-tasks<0..*>;]
} ma-capability-obj;
```

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

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

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

   ma-capability-version: The version of the measurement agent.

   ma-capability-tags: An optional unordered set of tags that
                             provide additional information about the
                             capabilities of the measurement agent.

   ma-capability-tasks: An optional unordered set of capability
                             objects for each supported task.

3.5.2. Definition of ma-capability-task-obj

```
   object {
       string              ma-capability-task-name;
       ma-registry-obj     ma-capability-task-functions<0..*>;
       string              ma-capability-task-version;
   } ma-capability-task-obj;
```

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

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

   ma-capability-task-functions: A possibly empty unordered set of
                                 registry entries identifying
                                 functions this task implements.

   ma-capability-task-version: The version of the measurement task.

3.5.3. Definition of ma-status-obj

```
   object {
      uuid                      ma-status-agent-id;
      [uri                      ma-status-device-id;]
       datetime                 ma-status-last-started;
       ma-status-interface-obj  ma-status-interfaces<0..*>;
      [ma-status-schedule-obj   ma-status-schedules<0..*>;]
      [ma-status-suppression-obj  ma-status-suppressions<0..*>;]
   } ma-status-obj;
```

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

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

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

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

    ma-status-interfaces: An unordered set of network interfaces
                               available on the device.

    ma-status-schedules: An optional unordered set of status objects
                               for each schedule.

    ma-status-suppressions: An optional unordered set of status objects
                               for each suppression.

3.5.4. Definition of ma-status-schedule-obj

    object {
        string   ma-status-schedule-name;
        string   ma-status-schedule-state;
        int   ma-status-schedule-storage;
        counter   ma-status-schedule-invocations;
        counter   ma-status-schedule-suppressions;
        counter   ma-status-schedule-overlaps;
        counter   ma-status-schedule-failures;
        datetime   ma-status-schedule-last-invocation;
       [ma-status-action-obj   ma-status-schedule-actions<0..*>;]
    } ma-status-schedule-obj;

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

    ma-status-schedule-name: The name of the schedule this
                                      status object refers to.

    ma-status-schedule-state: The state of the schedule. The
                                      value 'enabled' indicates that
                                      the schedule is currently
                                      enabled. The value 'suppressed'
                                      indicates that the schedule is
                                      currently suppressed. The value
                                      'disabled' indicates that the
                                      schedule is currently disabled.
                                      The value 'running' indicates
                                      that the schedule is currently
                                      running.

    ma-status-schedule-storage: The amount of secondary storage
                                      (e.g., allocated in a file

                                        system) holding temporary data
                                        allocated to the schedule in
                                        bytes. This object reports the
                                        amount of allocated physical
                                        storage and not the storage used
                                        by logical data records. Data
                                        models should use a 64-bit
                                        integer type.

   ma-status-schedule-invocations   Number of invocations of this
                                        schedule. This counter does not
                                        include suppressed invocations or
                                        invocations that were prevented
                                        due to an overlap with a previous
                                        invocation of this schedule.

   ma-status-schedule-suppressions   Number of suppressed executions
                                        of this schedule.

   ma-status-schedule-overlaps   Number of executions prevented
                                        due to overlaps with a previous
                                        invocation of this schedule.

   ma-status-schedule-failures   Number of failed executions of
                                        this schedule. A failed
                                        execution is an execution where
                                        at least one action failed.

   ma-status-schedule-last-invocation: The date and time of the last
                                        invocation of this schedule.

   ma-status-schedule-actions: An optional ordered list of
                                        status objects for each action of
                                        the schedule.

3.5.5. Definition of ma-status-action-obj

```
    object {
        string              ma-status-action-name;
        string              ma-status-action-state;
        int                 ma-status-action-storage;
        counter             ma-status-action-invocations;
        counter             ma-status-action-suppressions;
        counter             ma-status-action-overlaps;
        counter             ma-status-action-failures;
        datetime            ma-status-action-last-invocation;
        datetime            ma-status-action-last-completion;
        int                 ma-status-action-last-status;
        string              ma-status-action-last-message;
        datetime            ma-status-action-last-failed-completion;
        int                 ma-status-action-last-failed-status;
        string              ma-status-action-last-failed-message;
    } ma-status-action-obj;
```

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

ma-status-action-name: The name of the action of a
                                        schedule this status object
                                        refers to.

ma-status-action-state: The state of the action.
                                        The value 'enabled'
                                        indicates that the action is
                                        currently enabled. The
                                        value 'suppressed' indicates
                                        that the action is currently
                                        suppressed. The value
                                        'disabled' indicates that
                                        the action is currently
                                        disabled. The value
                                        'running' indicates that the
                                        action is currently running.

ma-status-action-storage: The amount of secondary
                                        storage (e.g., allocated in
                                        a file system) holding
                                        temporary data allocated to
                                        the action in bytes. This
                                        object reports the amount of
                                        allocated physical storage
                                        and not the storage used by
                                        logical data records. Data
                                        models should use a 64-bit
                                        integer type.

```
ma-status-action-invocations           Number of invocations of
                                       this action.  This counter
                                       does not include suppressed
                                       invocations or invocations
                                       that were prevented due to
                                       an overlap with a previous
                                       invocation of this action.

ma-status-action-suppressions          Number of suppressed
                                       executions of this action.

ma-status-action-overlaps              Number of executions
                                       prevented due to overlaps
                                       with a previous invocation
                                       of this action.

ma-status-action-failures              Number of failed executions
                                       of this action.

ma-status-action-last-invocation:      The date and time of the
                                       last invocation of this
                                       action.

ma-status-action-last-completion:      The date and time of the
                                       last completion of this
                                       action.

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

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

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

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

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

3.5.6. Definition of ma-status-suppression-obj

```
object {
    string                  ma-status-suppression-name;
    string                  ma-status-suppression-state;
} ma-status-suppression-obj;
```

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

ma-status-suppression-name: The name of the suppression this status
                              object refers to.

ma-status-suppression-state: The state of the suppression. The
                              value 'enabled' indicates that the
                              suppression is currently enabled. The
                              value 'active indicates that the
                              suppression is currently active. The
                              value 'disabled' indicates that the
                              suppression is currently disabled.

3.5.7. Definition of ma-status-interface-obj

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

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

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

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

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

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

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

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

    ma-status-interface-dns-servers:        An optional ordered 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 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 Action 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. It is generally suggested that
implementations running out of storage stop executing new measurement
tasks and retain old measurement data.

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 Action. The Action
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.

Whereas 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;]
       [string                  ma-report-measurement-point;]
       [ma-report-result-obj  ma-report-results<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-measurement-point: An optional identifier for the
                               measurement point indicating where the
                               measurement agent is located on a path
                               (see [RFC7398] for further details).

   ma-report-results: An optional and possibly empty
                               unordered set of result objects.

3.6.2. Definition of ma-report-result-obj

```
   object {
       string                 ma-report-result-schedule-name;
       string                 ma-report-result-action-name;
       string                 ma-report-result-task-name;
      [ma-option-obj          ma-report-result-options<0..*>;]
      [string                 ma-report-result-tags<0..*>;]
       datetime               ma-report-result-event-time;
       datetime               ma-report-result-start-time;
      [datetime               ma-report-result-end-time;]
      [string                 ma-report-result-cycle-number;]
       int                    ma-report-result-status;
      [ma-report-conflict-obj ma-report-result-conflicts<0..*>;]
      [ma-report-table-obj    ma-report-result-tables<0..*>;]
   } ma-report-result-obj;
```

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

   ma-report-result-schedule-name: The name of the schedule that
                                    produced the result.

   ma-report-result-action-name: The name of the action in the
                                    schedule that produced the result.

   ma-report-result-task-name: The name of the task that produced
                                    the result.

   ma-report-result-options: An optional ordered joined list of
                                    options provided by the task object
                                    and the action object when the action
                                    was started.

   ma-report-result-tags: An optional unordered set of tags.
                                    This is the joined set of tags
                                    provided by the task object and the
                                    action object and schedule object
                                    when the action was started.

   ma-report-result-event-time: The date and time of the event that
                                    triggered the schedule of the action
                                    that produced the reported result
                                    values. The date and time does not
                                    include any added randomization.

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

ma-report-result-end-time: An optional date and time indicating
                                  when the action finished.

ma-report-result-cycle-number: An optional cycle number derived from
                                  ma-report-result-event-time. It is
                                  the time closest to ma-report-result-
                                  event-time that is a multiple of the
                                  ma-event-cycle-interval of the event
                                  that triggered the execution of the
                                  schedule. The value is only present
                                  in an ma-report-result-obj if the
                                  event that triggered the execution of
                                  the schedule has a defined ma-event-
                                  cycle-interval. The cycle number is
                                  represented in the format
                                  YYYYMMDD.HHMMSS where YYYY represents
                                  the year, MM the month (1..12), DD
                                  the day of the months (01..31), HH
                                  the hour (00..23), MM the minute
                                  (00..59), and SS the second (00..59).
                                  The cycle number is using Coordinated
                                  Universal Time (UTC).

ma-report-result-status: The status code returned by the
                                  execution of the action.

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

ma-report-result-tables: An optional and possibly empty
                                  unordered set of result tables.

3.6.3. Definition of ma-report-conflict-obj

```
object {
    string  ma-report-conflict-schedule-name;
    string  ma-report-conflict-action-name;
    string  ma-report-conflict-task-name;
} ma-report-conflict-obj;
```

The ma-report-conflict-obj provides the information about conflicting
action that might have impacted the measurement results. It consists
of the following elements:

ma-report-result-schedule-name: The name of the schedule that may
                                have impacted the result.

```
   ma-report-result-action-name:   The name of the action in the
                                   schedule that may have impacted the
                                   result.

   ma-report-result-task-name:     The name of the task that may have
                                   impacted the result.
```

3.6.4. Definition of ma-report-table-obj

```
   object {
       [ma-registry-obj    ma-report-table-functions<0..*>;]
       [string]            ma-report-table-column-labels<0..*>;]
       [ma-report-row-obj  ma-report-table-rows<0..*>;]
   } ma-report-table-obj;
```

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

```
   ma-report-table-functions:      An optional and possibly empty
                                   unordered set of registry entries
                                   identifying the functions for which
                                   results that are reported.

   ma-report-table-column-labels:  An optional and possibly empty
                                   ordered list of column labels.

   ma-report-table-rows:           A possibly empty ordered list of
                                   result rows.
```

3.6.5. Definition of ma-report-row-obj

```
   object {
        data               ma-report-row-values<0..*>;
   } ma-report-row-obj;
```

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

```
   ma-report-row-values:    A possibly empty ordered list of result
                            values.  When present, it contains an
                            ordered list of values that align to the
                            set of column labels for the report.
```

3.7. Common Objects: Schedules

   A Schedule specifies the execution of a single or repeated series of
   Actions. An Action extends a Configured Task with additional
   specific parameters. Each Schedule contains basically two elements:

an ordered list of Actions to be executed and an Event object
triggering the execution of the Schedule. The Schedule states what
Actions to run (with what configuration) and when to run the Actions.
A Schedule may optionally have an Event that stops the execution of
the Schedule or a maximum duration after which a schedule is stopped.

Multiple Actions contained as an ordered list of a single Measurement
Schedule will be executed according to the execution mode of the
Schedule. In sequential mode, Actions will be executed sequentially
and in parallel mode, all Actions will be executed concurrently. In
pipelined mode, data produced by one Action is passed to the
subsequent Action. Actions contained in different Schedules execute
in parallel with such conflicts being reported in the Reporting
Information where necessary. 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 Actions to execute, the Schedule also
specifies how to link the data outputs from each Action to other
Schedules. 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. A single Task producing multiple different outputs is
expected to properly tag the different result. An Action receiving
the output can then filter the results based on the tag if necessary.
For example, a Measurement Task might report routine results to a
data Reporting Task in a Schedule that communicates hourly via the
Broadband PPP interface, but also outputs emergency conditions via an
alarm Reporting Task in a different Schedule 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 Schedules
or Actions, this Information Model does not mandate or define a
queue. The Information Model, however, reports the storage allocated
to Schedules and Actions so that storage usage can be monitored.
Furthermore, it is recommended that MA implementations by default
retain old data and stop the execution of new measurement tasks if
the MA runs out of storage capacity.

When specifying the task to execute within the Schedule, i.e.,
creating an Action, it is possible to add to the Action 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 as option parameters of the
Action in 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,
Action 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 Action and the Task
Configuration, the Action Options are appended to those specified in
the Task Configuration.

Example: An Action of a Schedule references a single Measurement
   Task Configuration for measuring UDP latency. It specifies that
   results are to be sent to a Schedule with a Reporting Action.
   This Reporting Task of the Reporting Action is executed by a
   separate Schedule that specifies that it should run hourly at 5
   minutes past the hour. When run this Reporting Action takes the
   data generated by the UDP latency Measurement 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.

Schedules and Actions may optionally also be given tags that are
included in result reports sent to a Collector. In addition,
schedules can be given suppression tags that may be used to select
Schedules and Actions for suppression.

3.7.1. Definition of ma-schedule-obj

```
object {
    string              ma-schedule-name;
    ma-event-obj        ma-schedule-start;
   [ma-event-obj        ma-schedule-end;]
   [int                 ma-schedule-duration;]
    ma-action-obj       ma-schedule-actions<0..*>;
    string              ma-schedule-execution-mode;
   [string              ma-schedule-tags<0..*>;]
   [string              ma-schedule-suppression-tags<0..*>;]
} 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-start: An event object indicating when the
                              schedule starts.

ma-schedule-end: An optional event object controlling
                              the forceful termination of scheduled
                              actions. When the event occurs, all
                              actions of the schedule will be forced
                              to terminate gracefully.

ma-schedule-duration: An optional duration in seconds for the
                              schedule. All actions of the schedule
                              will be forced to terminate gracefully
                              after the duration number of seconds
                              past the start of the schedule.

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

ma-schedule-execution-mode: Indicates whether the actions should be
                              executed sequentially, in parallel, or
                              in a pipelined mode (where data
                              produced by one action is passed to the
                              subsequent action). The default
                              execution mode is pipelined.

ma-schedule-tags: An optional unordered set of tags that
                              are reported together with the
                              measurement results to a collector.

ma-schedule-suppression-tags: An optional unordered set of
                              suppression tags that are used to
                              select schedules to be suppressed.

3.7.2. Definition of ma-action-obj

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

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

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

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

ma-action-task-options: An optional and possibly empty ordered
                              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
                              unordered set of names of destination
                              schedules that consume output produced
                              by this action.

ma-action-tags: An optional unordered set of tags that
                              are reported together with the
                              measurement results to a collector.

ma-action-suppression-tags: An optional unordered set of
                              suppression tags that are used to
                              select actions to be suppressed.

3.8. Common Objects: Channels

   A Channel defines a bi-directional communication mechanism 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 a 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 its 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 used for reporting results may specify that
      results are to be sent to the URL (https://collector.example.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 IP
                               protocol stack 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 may have registry entries 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. Registry entries are 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, these URIs 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 may also refer to registry
entries 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.

An 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, defined as an
Option, to indicate whether to send a report without measurement
results 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.

In addition the Task Configuration may optionally also be given tags
that can carry 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;
       ma-registry-obj     ma-task-functions<0..*>;
      [ma-option-obj       ma-task-options<0..*>;]
      [string              ma-task-tags<0..*>;]
   } ma-task-obj;
```

The ma-task-obj defines a configured task that can be invoked as part
of an action. A configured task can be referenced by its name and it
contains a possibly empty set of URIs to link to registry entries.
Options allow the configuration of task parameters (in the form of
name-value pairs). The ma-task-obj consists of the following
elements:

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

ma-task-functions: A possibly empty unordered set of registry
                        entries identifying the functions of the
                        configured task.

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

ma-task-tags: An optional unordered set of tags that are
                        reported together with the measurement
                        results to a collector.

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.

The ma-option-obj is used to define Task Configuration Options. Task
Configuration Options are generally task specific. For tasks
associated with an entry in a registry, the registry may define well-
known option names (e.g., the so-called parameters in the IPPM metric
registry [I-D.ietf-ippm-metric-registry]). Control and Reporting
Tasks need to know the Channel they are going to use. The common
option name for specifying the channel is "channel" where the
option's value refers to the name of an ma-channel-obj.

3.10. Common Objects: Registry Information

Tasks and actions can be associated with entries in a registry. A
registry object refers to an entry in a registry (identified by a
URI) and it may define a set of roles.

3.10.1. Definition of ma-registry-obj

```
object {
    uri                 ma-registry-uri;
   [string              ma-registry-role<0..*>;]
} ma-registry-obj;
```

The ma-registry-obj refers to an entry of a registry and it defines
the associated role(s). The ma-registry-obj consists of the
following elements:

ma-registry-uri: A URI identifying an entry in a registry.

ma-registry-role: An optional and possibly empty unordered
                         set of roles for the identified registry
                         entry.

3.11. Common Objects: Event Information

The Event information object used throughout the information models
can initially take one of several different forms. Additional forms
may be defined later in order to bind the execution of schedules to
additional events. The initially defined Event forms are:

1. Periodic Timing: Emits multiple events periodically according to
    an interval time defined in seconds

   2. Calendar Timing: Emits multiple events according to a calendar
       based pattern, e.g., 22 minutes past each hour of the day on
       weekdays

   3. One Off Timing: Emits one event at a specific date and time

   4. Immediate: Emits one event as soon as possible

   5. Startup: Emits an event whenever the MA is started (e.g., at
       device startup)

   6. Controller Lost: Emits an event when connectivity to the
       controller has been lost

   7. Controller Connected: Emits an event when connectivity to the
       controller has been (re-)established

   Optionally each of the Event 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
   seconds over which the start of the task is delayed from the starting
   times specified by the event object.

   Both the Periodic and Calendar timing objects allow for a series of
   Actions 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.

   Startup events are only created 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].

3.11.1. Definition of ma-event-obj

```
   object {
      string                 ma-event-name;
      union {
         ma-periodic-obj           ma-event-periodic;
         ma-calendar-obj           ma-event-calendar;
         ma-one-off-obj            ma-event-one-off;
         ma-immediate-obj          ma-event-immediate;
         ma-startup-obj            ma-event-startup;
         ma-controller-lost-obj    ma-event-controller-lost;
         ma-controller-connected-obj ma-event-controller-connected;
      }
      [int                   ma-event-random-spread;]
      [int                   ma-event-cycle-interval;]
   } ma-event-obj;
```

The ma-event-obj is the main event object. Event objects are
identified by a name. A generic event object itself contains a more
specific event object. The set of specific event objects should be
extensible. The initial set of specific event objects is further
described below. The ma-event-obj also includes an optional uniform
random spread that can be used to randomize the start times of
schedules triggered by an event. The ma-event-obj consists of the
following elements:

ma-event-name: The name uniquely identifies an event
                                    object. Schedules refer to event
                                    objects by this name.

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

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

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

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

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

ma-event-controller-lost: The ma-event-controller-lost is
                                    present for connectivity to
                                    controller lost event objects.

```
ma-event-controller-connected:  The ma-event-controller-connected is
                                present for connectivity to a
                                controller established event objects.

ma-event-random-spread:         The optional ma-event-random-spread
                                adds a random delay defined in
                                seconds to the event object.  No
                                random delay is added if ma-event-
                                random-spread does not exist.

ma-event-cycle-interval:        The optional ma-event-cycle-interval
                                defines the duration of the time
                                interval in seconds that is used to
                                calculate cycle numbers.  No cycle
                                number is calculated if ma-event-
                                cycle-interval does not exist.
```

3.11.2. 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. Schedules 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
                         Schedules using this object are first
                         started.  If not present it defaults to
                         immediate.

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

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

3.11.3. Definition of ma-calendar-obj

Calendar Timing supports the routine execution of Schedules at
specific times and/or on specific dates. It can support more
flexible timing than Periodic Timing since the execution of Schedules

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 times 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
                            Schedules using this object are first
                            started. If not present it defaults to
                            immediate.

ma-calendar-end: The optional date and time at which
                            Schedules using this object are 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 the week.
                                  If not present, it defaults to no days.

      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.

3.11.4. 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.
   Schedules 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 Schedules using
                               this object are started.
```

3.11.5. Definition of ma-immediate-obj

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

   The ma-immediate-obj event object has no further information
   elements. Schedules using an ma-immediate-obj are started as soon as
   possible.

3.11.6. Definition of ma-startup-obj

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

   The ma-startup-obj event object has no further information elements.
   Schedules or suppressions using an ma-startup-obj are started at MA
   initialization time.

3.11.7. Definition of ma-controller-lost-obj

```
   object {
                           // empty
   } ma-controller-lost-obj;
```

   The ma-controller-lost-obj event object has no further information
   elements. The ma-controller-lost-obj indicates that connectivity to
   the controller has been lost. This is determined by a timer started
   after each successful contact with a controller. When the timer
   reaches the controller-timeout (measured in seconds), an ma-
   controller-lost-obj event is generated. This event may be used to
   start a suppression.

3.11.8. Definition of ma-controller-connected-obj

```
   object {
                           // empty
   } ma-controller-connected-obj;
```

   The ma-controller-connected-obj event object has no further
   information elements. The ma-controller-connected-obj indicates that
   connectivity to the controller has been established again after it
   was lost. This event may be used to end a suppression.

4. Example Execution

   The example execution has two event sources E1 and E2 and three
   schedules S1, S2, and S3. The schedule S3 is started by events of
   event source E2 while the schedules S1 and S2 are both started by
   events of the event source E1. The schedules S1 and S2 have two
   actions each and schedule S3 has a single action. The event source
   E2 has no randomization while the event source E1 has the
   randomization r.

   Figure 2 shows a possible timeline of an execution. The time T is
   progressing downwards. The dotted vertial line indicates progress of
   time while a dotted horizontal line indicates which schedule are
   triggered by an event. Tilded lines indicate data flowing from an
   action to another schedule. Actions within a schedule are named A1,
   A2, etc.

```
       E2   E1   T            S1             S2            S3
                          sequential     parallel      pipelined
                  :
            e0 +
                  :
                  :
           e0+r +  .......... + .......... ++
                  :           | A1      A1 || A2
                  :           +           |+ ~~~~~~~>
                  :           | A2        |
                  :           |           + ~~~~~~~>
                  :           + ~~~~~~~~~~~~~~~~~~~~~>
                  :
                  :
            e1 +
                  :
           e1+r +  .......... + .......... ++
                  :           | A1      A1 ||
                  :           |           +|~~~~~~~>
                  :           |           | A2
                  :           +           +~~~~~~~>
                  :           | A2
                  :           + ~~~~~~~~~~~~~~~~~~~>
        e0     +  .................................... +
                  :                                    | A1
            e3 +                                       |
           e3+r +  .......... + .......... ++          |
                  :           | A1      A1 || A2        |
                  :           +           ++ ~~~~~>     |
                  :           | A2                      +
                  :           + ~~~~~~~~~~~~~~~~~~~>
                  V
```

Figure 2: Example Execution
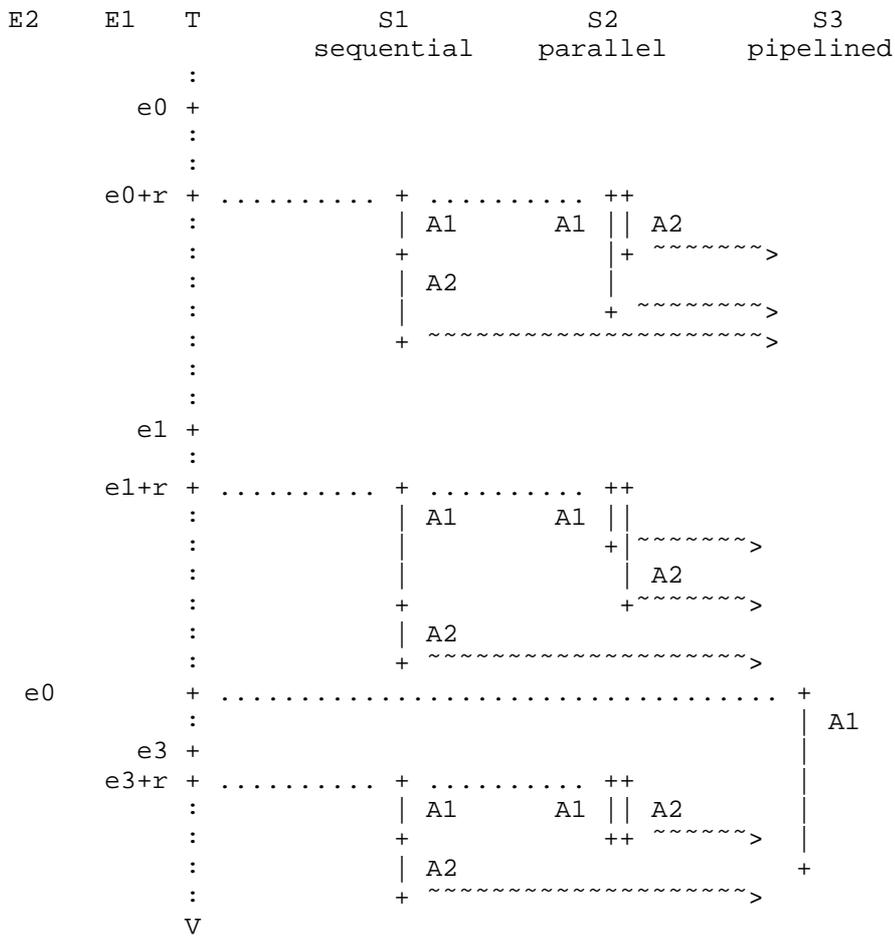
Note that implementations must handle possible concurrency issues.
In the example execution, action A1 of schedule S3 is consuming the
data that has been forwarded to schedule S3 while additional data is
arriving from action A2 of schedule S2.

5. IANA Considerations

This document makes no request of IANA.

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

6. 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
   Informaion 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 distributed
   denial of service 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 may, 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.
   Implementations and deployments should also be careful about exposing
   device-ids when this is not strictly needed.

   An implementation of this Information Model should support all the
   security and privacy requirements associated with the LMAP Framework
   [RFC7594]. In addition, users of this Information Model are advised
   to choose identifiers for Group IDs, tags or names of information
   model objects (e.g., configured tasks, schedules or actions) that do
   not reveal any sensitive information to people authorized to process
   measurement results but who are not authorized to know details about
   the Measurement Agents that were used to perform the measurement.

7. Acknowledgements

   Several people contributed to this specification by reviewing early
   versions and actively participating in the LMAP working group
   (apologies to those unintentionally omitted): Vaibhav Bajpai, Michael
   Bugenhagen, Timothy Carey, Alissa Cooper, Kenneth Ko, Al Morton, Dan
   Romascanu, Henning Schulzrinne, Andrea Soppera, Barbara Stark, and
   Jason Weil.

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

8. References

8.1. Normative References

   [ISO.10646]
              International Organization for Standardization,
              "Information Technology - Universal Multiple-Octet Coded
              Character Set (UCS)", ISO Standard 10646:2014, 2014.

   [POSIX.2] The IEEE and The Open Group, "The Open Group Base
              Specifications Issue 7", IEEE Standard 1003.1-2008, 2016.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/
              RFC2119, March 1997,
              <http://www.rfc-editor.org/info/rfc2119>.

   [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet:
              Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002,
              <http://www.rfc-editor.org/info/rfc3339>.

   [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
              Resource Identifier (URI): Generic Syntax", STD 66, RFC
              3986, DOI 10.17487/RFC3986, January 2005,
              <http://www.rfc-editor.org/info/rfc3986>.

   [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally
              Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10
              .17487/RFC4122, July 2005,
              <http://www.rfc-editor.org/info/rfc4122>.

8.2. 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-10 (work in progress), November 2016.

   [I-D.ietf-lmap-yang]
              Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for
              LMAP Measurement Agents", draft-ietf-lmap-yang-10 (work in
              progress), January 2017.

   [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between
              Information Models and Data Models", RFC 3444, DOI 10
              .17487/RFC3444, January 2003,
              <http://www.rfc-editor.org/info/rfc3444>.

   [RFC7398] Bagnulo, M., Burbridge, T., Crawford, S., Eardley, P., and
              A. Morton, "A Reference Path and Measurement Points for
              Large-Scale Measurement of Broadband Performance", RFC
              7398, DOI 10.17487/RFC7398, February 2015,
              <http://www.rfc-editor.org/info/rfc7398>.

   [RFC7536] Linsner, M., Eardley, P., Burbridge, T., and F. Sorensen,
              "Large-Scale Broadband Measurement Use Cases", RFC 7536,
              DOI 10.17487/RFC7536, May 2015,
              <http://www.rfc-editor.org/info/rfc7536>.

   [RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T.,
              Aitken, P., and A. Akhter, "A Framework for Large-Scale
              Measurement of Broadband Performance (LMAP)", RFC 7594,
              DOI 10.17487/RFC7594, September 2015,
              <http://www.rfc-editor.org/info/rfc7594>.

Appendix A.   Change History

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

A.1. Non-editorial changes since -17

   o The information model is subdivided into aspects and not sections.

   o Changes to address the GEN-ART review comments.

A.2. Non-editorial changes since -16

   o Addressing Alissa Cooper's review comments.

A.3. Non-editorial changes since -15

   o The reference to the framework is now informational.

A.4. Non-editorial changes since -14

   o Clarified that the cycle number is in UTC.

A.5. Non-editorial changes since -13

   o Removed the ma-config-device-id from the ma-config-obj.

   o Added ma-config-report-group-id and clarified how two flags ma-
      config-report-agent-id and ma-config-report-group-id work.

A.6. Non-editorial changes since -12

   o Renamed the ma-metrics-registry-obj to ma-registry-obj since tasks
      may refer to different registries (not just a metrics registry).

   o Clarifications and bug fixes.

A.7. Non-editorial changes since -11

   o Clarifications and bug fixes.

A.8. Non-editorial changes since -10

   o Rewrote the text concerning the well-known "channel" option name.

   o Added ma-report-result-event-time, ma-report-result-cycle-number,
      and ma-event-cycle-interval.

   o Added ma-capability-tags.

   o Added a new section showing an example execution.

   o Several clarifications and bug fixes.

A.9. Non-editorial changes since -09

   o Added ma-status-schedule-storage and ma-status-action-storage.

   o Removed suppress-by-default.

   o Moved ma-report-result-metrics of the ma-report-result-obj to ma-
      report-table-metrics of the ma-report-table-obj so that the
      relationship between metrics and result tables is clear.

   o Added ma-report-conflict-obj.

   o Added ma-report-result-status to ma-report-result-obj.

   o Several clarifications and bug fixes.

A.10. Non-editorial changes since -08

   o Refactored the ma-report-task-obj into the ma-report-result-obj.

   o Introduced the ma-report-table-obj so that a result can contain
      multiple tables.

   o Report schedule, action, and task name as part of the ma-report-
      result-obj.

   o Report conflicts per ma-report-result-obj and not per ma-report-
      row-obj.

   o Report the start/end time as part of the ma-report-result-obj.

A.11. Non-editorial changes since -07

   o Added ma-schedule-end and ma-schedule-duration.

   o Changed the granularity of scheduler timings to seconds.

   o Added ma-status-suppression-obj to report the status of
      suppressions as done in the YANG data model.

   o Added counters to schedule and action status objects to match the
      counters in the YANG data model.

   o Using tags to pass information such as a measurement cycle
      identifier to the collector.

   o Using suppression tags and glob-style matching to select schedules
      and actions to be suppressed.

A.12. Non-editorial changes since -06

   o The default execution mode is pipelined (LI12)

   o Added text to define which action consumes data in sequential,
      pipelines, and parallel execution mode (LI11)

   o Added ma-config-measurement-point, ma-report-measurement-point,
      and ma-config-report-measurement-point to configure and report the
      measurement point (LI10)

   o Turned ma-suppression-obj into a list that uses a start event and
      a stop event to define the start and end of suppression; this
      unifies the handling of suppression and loss of controller
      connectivity (LI09)

o Added ma-controller-lost-obj and ma-controller-ok-obj event
  objects (LI09)

o Added ma-status-schedule-obj to report the status of a schedule
  and refactored ma-task-status-obj into ma-status-action-obj to
  report the status of an action (LI07, LI08)

o Introduced a common ma-metric-registry-obj that identifies a
  metric and a set of associated roles and added this object to
  expose metric capabilities and to support the configuration of
  metrics and to report the metrics used (LI06)

o Introduced ma-capability-obj and ma-capability-task-obj to expose
  the capabilities of a measurement agent (LI05)

o Use 'ordered list' or 'unordered set' instead of list, collection,
  etc. (LI02)

o Clarification that Actions are part of a Schedule (LI03)

o Deleted terms that are not strictly needed (LI04)

A.13. Non-editorial changes since -05

o A task can now reference multiply registry entries.

o Consistent usage of the term Action and Task.

o Schedules are triggered by Events instead of Timings; Timings are
  just one of many possible event sources.

o Actions feed into other Schedules (instead of Actions within other
  Schedules).

o Removed the notion of multiple task outputs.

o Support for sequential, parallel, and pipelined execution of
  Actions.

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

                     Router Buffer Sizes In The WAN
                draft-ksubram-lmap-router-buffer-sizes-01

Abstract

   This draft identifies the set of data that needs to be collected, and
   analyzed to quantify router buffer sizes used in routers in the Wide
   Area Network (WAN). The scope of this draft is limited to WAN links
   that have link latencies of 40 to 150 milliseconds.

   Reducing router buffer sizes has many advantages, the most important
   being cost. However, there is not much data available today to
   effectively calculate this. This draft details use cases for the
   study, and lists data that needs to be taken into consideration to be
   able to quantify the size of router buffers. The details of the
   individual measurement metrics are beyond the scope of this document.
   Neither does the draft identify methods to gather the data. What it
   identifies is a need to be able to collect, and report this empirical
   data in a readable fashion thus providing the ability to study and
   compare data in a more standardized method.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   "How much buffering do core links need?" is a question that has been
   under study for a while. The question boils down to quantify buffer
   sizes and yet achieve 100% utilization on links with maximum
   throughput at a feasible cost.

   Buffer design could substantially increase costs. While over-
   buffering seems intuitive it can complicate the design of high speed
   routers, lead to higher power consumption, more board space, and
   lower density. It can actually increase end-to-end delay in the
   presence of congestion. This can make congestion more persistent.
   Additionally, there is always a tradeoff between buffer sizes and the
   capacity of a router.

   On the other hand, under-buffering while doing away from the above
   cons of over-buffering could lead us away from our primary goal of
   100 percent link utilization. This could happen in a scenario using
   a simple Additive Increase Multiplicative Decrease (AIMD) for TCP
   flows when the sender has packets to send but the window size
   advertised is less and as a result the receiver consumes far less
   that it could.

   The rule of thumb for router buffers has been defined as [Villamizar]
   : B = RTT*C. Where B, was the buffer size, RTT the Round Trip Time,
   and C the capacity of the bottleneck link. [RFC3429] also talks
   about the buffer size being at least one TCP window size.

   However later studies [Appenzeller], show that the rule of thumb
   works either for a single flow or a perfectly synchronized large
   number of flows. Further they postulate that the buffer size is
   actually (2RTT * C)/sqrt(n), where n is the number of flows. This
   indicates a significant reduction in the buffer chip promoting lower
   costs.

   As seen, there have been proponents for large buffers and small.
   However, most of these studies are based on theoretical models and
   simulations. Today, there is no model or protocol to mine big data
   from a providers network to be able to answer this question
   efficiently. The nature of WAN traffic can be uncertain and varying.
   Furthermore the traffic could vastly vary between individual ISPs.
   This document implored the need for a model of mining empirical big
   data in a providers network to be able to build a network that drives
   down the $/GB and at the same time maximizing link utilization.

   This document outlines use cases for the study of router buffer sizes
   in the WAN and identifies the data that needs to be collected and

analyzed. It could be further extended to the edge and datacenters,
but it is outside the scope of this draft.

2. Terminology

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].

3. Use Case

From an operator's perspective it is imperative to monitor discards
and link utilization over WAN links to be able to study the router
buffer sizes. But these alone will be unable to provide an operator
with enough information as to why the discards happened. The two use
cases outlined here argue that more data needs to be collected,
reported, and analyzed.

3.1. Discards with small buffer sizes

Trans-pacific and trans-atlantic links of latencies in the range of
150 ms and 90 ms respectively, with low link utilization of 30
percent, and small buffers have seen dropped packets. The most
intuitive method has been to increase the buffer sizes for these
links on noticing packet discards. While this might alleviate the
issue temporarily, unless the right problem has been identified this
could readily lead to buffer bloat which has many issues on its own.

3.2. Discards with large buffer sizes

Operators have also observed dropped packets on WAN links within
North America with as large buffers as 125 MB per port with link
utilizations of 60%. If this happens even if the router has not been
specifically configured to drop certain type of packets, or there are
no routing misconfigurations, then clearly the issue here is not the
size of the router buffer.

4. List of required data for study of router buffer sizes

This section talks about the absolute minimum requirements of the
type of data that needs to be collected to be able to effectively
quantify router buffer size.

		Data	Details
1		Number of concurrent flows, N	For aggregate traffic
2		Length of the flow, L	[Flow start time - flow end time]
3		Packet Discards, P	Per Interface
4		Reason for Packet Discards, R	Buffer overflow, configuration, etc.
5		Resolution of Time Interval, T	[Flow start time - flow end time]
6		5 tuple flow identity, I	Src IP, Dest IP, Src port, Dest Port, Protocol.

Table 1: List of required data for Router Buffer Sizes

A service provider needs to take into consideration several
attributes to determine the right buffer size for its WAN routers.
This section enlists the details as to why the five above have been
identified as the minimum essential data needed to aid the study of
router buffer sizes.

4.1. Number of concurrent flows, N

Studies [Feldmann] and [Stevens] show that 95% of flows in the
internet today are attributed to TCP [Postel] flows. The nature of
these flows can vary significantly not only with various time
periods, but also between providers. Flows that spend most of their
time in slow-start require significantly less buffering than flows
that live mostly in congestion avoidance. Due to this it is
important to identify the type of concurrent flows that can live on a
WAN link.

Short (non-persistent) flows are those that live for less than one
RTT, and large (persistent) flows are those whose lifetime is larger
that one RTT with congestion overhead. Internet measurements [Avra]
show that while a smaller number of large flows contribute to maximum
packet transfer, short flows dominate most TCP sessions and large
flows are known to have a larger effect on buffer sizes. These
combination flows could in turn have an effect on Round Trip Time
(RTT), loss probability and flow lengths. The ability to detect
large flows is necessary because while the flows can be constant in
steady state, the aggregate traffic can keep changing due to various
arrival and departure rates. There needs to be a way for the number
of concurrent flows to be collected and analyzed with the granularity
of the lifetime of short flows, as low as one millisecond.

4.2. Length of a flow, L

   Length of a flow can be defined as its duration: [flow stop time -
   flow start time], or the number of packets/bytes sent in this time
   duration. Identifying the length of flow in a provider's network
   will give information of the mix of short and large flows that are
   present in the WAN. This will lead to modeling implications in TCP
   flow control.

4.3. Packet Discards, D

   Number of packet discards per interface is probably the most
   important metric. Of this the number of outward (WAN) facing
   interface discards would be more intuitive to the study of buffer
   sizes. Interface discards can be referred to in [RFC2893]

4.4. Reason for Packet Discards, R

   There can be several reasons for packet discards especially when it
   is observed on less utilized links. Some of them could be due to
   routing misconfigurations, or designed to drop certain packets due to
   configurations. Clearly stating a reason as insufficient buffer will
   help narrow down the data required. This is especially true in the
   case of smart buffer allocations when some ports run out of buffers
   but not others. We could observe that a port has been allocated
   only, say, 30 percent of the available total buffer space but is
   experiencing the highest utilization and as a result of that is
   seeing packet drops pointing to the fact that dynamic buffers' smart
   allocations scheme is not adaptive and predictive to the nature of
   the WAN traffic.

4.5. Resolution of time interval, T

   The time interval should be granular such that it captures not only
   the number of concurrent flows in steady state but also the aggregate
   traffic over the lifetime of a short flow. It should also be able to
   correlate the discards per interface to the number of concurrent
   flows.

   Today via IPFIX we can calculate the number of concurrent flows. Via
   Sflow counters or flows, we can calculate the discards. Using
   counters requires upto two times the granularity set for any changes
   to be visible due to Nyquist rate. Reducing the counter export
   interval would increase the responsiveness, but at the cost of
   increased overhead and reduced scalability. On the other hand,
   packet sampling automatically allocates monitoring resources to busy
   links, providing a highly scaleable way to quickly detect traffic

flows wherever they occur in the network. Responsiveness is
important for a more stable control.

4.6. 5 Tuple Flow Identity, I

5 tuple flows have a source IP, destination IP, source port,
destination port, and protocol to identify endpoints for
unidirectional flows. Having this functionality gives the network
operator a way to identify the offending flows, legitimate elephant
flows, and high priority flows which may happen at certain periods
during the day. Being able to separate traffic using the 5 tuple,
further increases the strength of the sample set of empirical data
available for the study of router buffer sizes.

5. XML Representation of an Information Model for Calculating Router
    Buffer Sizes

The below is an example information model representing the data that
needs to be measured in order to make buffer size estimations easier.
The methods/algorithms to determine this date is out of the scope of
draft.

```
<?xml version="1.0" encoding="utf-8" ?>
<Router_Buffer_Schema>
  <Router>
    <!-- READ_ONLY -->
    <IPV4Address></IPV4Address>
    <Interface>
      <InterfaceID></InterfaceID>
      <InterfaceDirection></InterfaceDirection>
      <Capacity></Capacity>
      <FlowInformation>
        <NumConFlows></NumConFlows>
        <LengthOfFlow></LengthOfFlow>
        <SamplingRate></SamplingRate>
        <FiveTupleFlowInfo>
          <IPProtocol></IPProtocol>
          <SrcIPAddress></SrcIPAddress>
          <DestIPAddress></DestIPAddress>
          <SrcPort></SrcPort>
          <DestPort></DestPort>
        </FiveTupleFlowInfo>
        <FlowEndReason>
          <IdleTimeout></IdleTimeout>
          <ForcedTimeout></ForcedTimeout>
          <LackOfResources></LackOfResources>
          <ForcedEnd></ForcedEnd>
```

```
          <EndOfFlowDetected></EndOfFlowDetected>
        </FlowEndReason>
      </FlowInformation>
      <PacketDiscards>
        <EgressIfDiscard></EgressIfDiscard>
        <IngressIfDiscard></IngressIfDiscard>
        <ReasonPacketDiscards>
          <BufferOverrun></BufferOverrun>
          <RouterMisConfig></RouterMisConfig>
          <NetworkCongestion></NetworkCongestion>
        </ReasonPacketDiscards>
        <SamplPcktDiscRate></SamplPcktDiscRate>
        <PktDiscardHistogram></PktDiscardHistogram>
      </PacketDiscards>
      <LinkUtil></LinkUtil>
      <RoundTripTime></RoundTripTime>
      <!-- READ_WRITE -->
      <EnableRtrBuffCal></EnableRtrBuffCal>
      <BufferSize>
        <Percentage></Percentage>
        <Latency></Latency>
        <SizeInBytes>
          <BytesTotal></BytesTotal>
          <BytesUsed></BytesUsed>
          <BytesAvail></BytesAvail>
        </SizeInBytes>
        <HistogramSamplingRate></HistogramSamplingRate>
        <BufferSizeHistogram></BufferSizeHistogram>
      </BufferSize>
    </Interface>
  </Router>
</Router_Buffer_Schema>
```

6. Conclusion

   We see that there are numerous issues at different layers that have
   an effect (directly or indirectly) on the sizing of router buffers.
   We also notice that there is no study that takes empirical data into
   consideration. Ideally, what would be required is an all knowing
   oracle that sees the traffic flow on an end-to-end network across all
   layers. Due to a lack of the resource, the first step to the study
   of router buffer sizes is to effectively mine the big data repository
   of a provider for the data identified in this draft.

7. Acknowledgements

8. IANA Considerations

    This memo includes no request to IANA.

    All drafts are required to have an IANA considerations section (see
    the update of RFC 2434 [I-D.narten-iana-considerations-rfc2434bis]
    for a guide). If the draft does not require IANA to do anything, the
    section contains an explicit statement that this is the case (as
    above). If there are no requirements for IANA, the section will be
    removed during conversion into an RFC by the RFC Editor.

9. Security Considerations

    This document does not introduce new security issues.

10. References

10.1. Normative References

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

10.2. Informative References

    [Appenzeller]
                G. Appenzeller, I. Klesassy, and N. McKeown, "Some
                Internet Architectural Guidelines and Philosophy", 2004,
                <SIGCOMM '04 Proceedings of the 2004 conference on
                Applications, technologies, architectures, and protocols
                for computer communications>.

    [Avra]   Konstantin Avrachenkov, INRIA Sophia Antipolis,
                "Differentiation Between Short and Long TCP Flows:
                Predictability of the Response Time", 2004, <IEEE
                INFOCOMM>.

    [Feldmann]
                A. Feldmann, J. Rexford, and R. Caceres, "Efficient
                policies for carrying Web traffic over flow-switched
                networks", Dec. 1998, <IEEE/ACM Trans. Networking, vol. 6,
                pp. 673-685>.

   [I-D.narten-iana-considerations-rfc2434bis]
             Narten, T. and H. Alvestrand, "Guidelines for Writing an
             IANA Considerations Section in RFCs", draft-narten-iana-
             considerations-rfc2434bis-09 (work in progress), March
             2008.

   [Postel]   J. Postel, "Transmission Control Protocol", Sep. 1981,
             <RFC 793>.

   [RFC2893] K. McCloghrie, F. Kastenholz, "The Interfaces Group MIB",
             Jun. 2000, <RFC 2893>.

   [RFC3429] R. Bush and D. Meyer, "Some Internet Architectural
             Guidelines and Philosophy", Dec. 2002, <RFC 3429>.

   [Stevens] W. R. Stevens, "Transmission Control Protocol", 1994,
             <TCP/IP Illustrated. Reading, MA: Addison-Wesley, vol. 1>.

   [Villamizar]
             C. Villamizar and C. Song, "High performance tcp in
             ansnet", 1994, <ACM Computer Communications Review,
             24(5):45-60>.

Appendix A. Additional Stuff

   This becomes an Appendix.

Authors' Addresses

   Kamala Subramaniam (editor)
   Microsoft
   Mountain View, CA 94043
   US

   Email: kasubra@microsoft.com

   Darren Loher
   Microsoft
   Redmond, WA 98052
   US

Network Working Group   V.Liu
Internet Draft   L.Deng
Intended status: Standards Track   ChinaMobile
Expires: November 7, 2015

                                                              D.Liu

                                                              S.Liu
                                                               CATR
                                                               C.Li
                                                        China Cache
                                                        May 6, 2015

                REST Style Large MeAsurement Platform Protocol
                          draft-liu-lmap-rest-03

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 November 6, 2015.

Abstract

This document defines and implements a LMAP protocol based on YANG
Model and Rest-style http for control and report in Large Scale
Performance Measurement (LMAP).

Table of Contents

1. Introduction

With the rapid development of Internet technology and the increasing complexity of broadband network architecture, it is quite difficult to do large scale network measurements due to the lack of the unified measurement system and cooperative protocols. Therefore, the Large-Scale Measurement of Broadband Performance (LMAP) working group is formed to standardize a large scale measurement system for performance measurements of all kinds of broadband access methods.

There are 3 main entities proposed in the LMAP architecture: [I-D.ietf-lmap-framework]

-The Measurement Agents (MAs): implemented in network to perform measurement tasks;

-Controller: responsible for creating and assigning the measurement tasks;

-Collector: in charge of collecting and storing measurement results.

This document presents REST-style protocols for LMAP based on the YANG model from [I-D.ietf-lmap-information-model].Section 2 defined terminology for REST-style LMAP protocol. Section 3 introduce the considerations on LMAP protocol along with the session's establishment sequence consideration. Section 4 and Section 5 discuss and introduce the design and implementation of the Control Protocol and Report Protocol based on REST-style.

2. Terminology

   This section defines terminology for REST-style LMAP protocol. Please
   note that defined terms are capitalized.

   -Data model: For control protocol and Report protocol, Data model is
   require for carrying instructions and report results between
   Controller-MA and MA-Collector. In this draft, the data model is
   defined in [I-D.ietf-lmap-information-model]

   Other terms used in this document are defined in [I-D.ietf-lmap-
   framework]. We include the definition of key component in this
   document which play important roles during implementation.

   -Bootstrap: A process that integrates a Measurement Agent into a
   Measurement System.

   -Channel: A bi-directional logical connection that is defined by
   specific Controller and MA, or Collector and MA, plus associated
   security.

   -Collector: A function that receives a Report from a Measurement
   Agent.

   -Controller: A function that provides a Measurement Agent with its
   Instruction.
   -Control Channel: A Channel between a Controller and a MA over which
   Instruction Messages and Capabilities, Failure and Logging
   Information are sent.

   -Control Protocol: The protocol delivering Instruction(s) from a
   Controller to a Measurement Agent. It also delivers Capabilities,
   Failure and Logging Information from the Measurement Agent to the
   Controller. It can also be used to update the MA's Configuration. It
   runs over the Control Channel.

   -Data Model: The implementation of an Information Model in a
   particular data modelling language [RFC3444].

   -Instruction: The description of Measurement Tasks for a MA to
   perform and the details of the Report for it to send. It is the
   collective description of the Measurement Task configurations, the
   configuration of the Measurement Schedules, the configuration of the
   Report Channel(s), the configuration of Report Schedule(s), and the
   details of any suppression.

-Measurement Agent (MA): The function that receives Instruction
Messages from a Controller and operates the Instruction by executing
Measurement Tasks (using protocols outside the initial LMAP work
scope and perhaps in concert with one or more other Measurement
Agents or Measurement Peers) and (if part of the Instruction) by
reporting Measurement Results to a Collector or Collectors.

-Measurement Peer (MP) :The function that assists a Measurement Agent
with Measurement Tasks and does not have an interface to the
Controller or Collector.

-Measurement Schedule: The schedule for performing Measurement Tasks.

-Measurement Task: The action performed by a particular Measurement
Agent that consists of the single assessment of a Metric through
operation of a Measurement Method role at a particular time, with all
of the role's Input Parameters set to specific values.

-Measurement Traffic: the packet(s) generated by some types of
Measurement Method that involve measuring some parameter associated
with the transfer of the packet(s).

-Report: The set of Measurement Results and other associated
information (as defined by the Instruction). The Report is sent by a
Measurement Agent to a Collector.

-Report Channel: A Channel between a Collector and a MA over which
Report messages are sent.

-Report Protocol: The protocol delivering Report(s) from a
Measurement Agent to a Collector. It runs over the Report Channel.

-Report Schedule: the schedule for sending Reports to a Collector.
-Subscriber: An entity (associated with one or more users) that is
engaged in a subscription with a service provider.

In this draft 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]. This document follows the standardized HTTP/1.1
[RFC7213] and JSON [RFC7159] specifications.

3. Protocol Consideration

   This section discuss considerations on design and the architecture of
   LAMP protocol along with some implementation consideration. Meantime,
   discuss the advantages of choose REST-Style in LMAP protocol
   implementation.

   3.1. Data Model overview

   YANG [RFC6020] is a data modeling language used to model
   configuration and state data manipulated by the Network Configuration
   Protocol (NETCONF) and YANG is used to model the operations and
   content layers of NETCONF. So we can use YANG to illustrate the data
   model of CP and RP. Some work can refer to [[I-D.ietf-lmap-yang-00]].

   3.2. Control Session Initiation Consideration

   A LMAP network should contains at least one Controller, one Collector
   and many Measurement Agents (MA). The control protocol is allowed the
   Controller to configure the MA with an Instruction to offer Measure
   Task, Job Schedule, and Results Report process. The MA receives the
   Control Instruction and act autonomously. Also, the Controller can
   update the MA's configuration by the using the Control Protocol.

   In the implementation of the control protocol, it SHOULD consider
   which side between Controller and MA should act to initiate the
   Control Session. It is feasible that the control session can be
   initiated from either Controller side or MA side. Meantime in this
   draft, it is proposed that the control session should be initiated
   from the Controller to MA based on the following reasons:

   a) If the initiation is from the Controller side, the session can be
   started almost instantly, without the considering the polling time of
   all the MAs, and making a lot effort to synchronization the start
   moment.

   b) If the initiation is from the MA side, when the session can be
   started is decided by the status of the Controller, such as high load
   or light load.

   c) Since the Controller initiates the session, it can periodically
   poll the MA to detect the activation state. However, when the
   corresponding MA locates behind a NAT/FW while the Controller is

located outside of the NAT/FW, the NAT/FW traversal techniques should
be employed such as STUN/TURN or port forwarding.

3.3. Report Session Initiation Consideration

According to the LMAP framework [draft-ietf-lmap-framework-13], the
report protocol is very simple: the MA sends a report and gets a ACK,
there is no more interaction between the MA and the Collector.It is
proposed that the report session can be initiated by the MA [RP-
MUST-1, draft-starkcarey-lmap-protocol-criteria-01], so the report
can be sent in real time when the report get ready by the MA. If the
session is initiated by the Collector, it must periodically poll the
MA thus increase the complexity of the protocol, so the way is not
recommended.

3.4. REST-Style LMAP Implement Consideration

This document follows the REST style in designing both the control
and report protocols for LMAP, for the following considerations:

   First, REST is much easier to be implemented the other methods
such as SOAP, most of open source web service developing packages
support REST, to deploy a REST application can be as easy as a few
lines of code [CP-DIFF-7, CP-DIFF-9, CP-DIFF-10, RP-DIFF-10, RP-DIFF-
11, draft-starkcarey-lmap-protocol-criteria-01]. Second, no special
requirement for the client, any browser or http client can visit the
REST application. Last, REST has clear resource annotation and finite
operations, all of those are readable, and easy to understand [CP-
MUST-4, RP-MUST-3, draft-starkcarey-lmap-protocol-criteria-01].

4. Control Protocol

This section discuss design and implementation of Control Protocol
based on REST style. The Control Protocol is being used between
Controller and MA to delivers Instruction Message. Control Protocol
SHOULD basically have three functions. First of all, it carries
instruct information from Controller to MA. Secondly, the Control
Protocol enable the MA to inform the Controller about its
Capabilities and Failure and Logging Information. Thirdly, carrying
update information from Controller to MA.

The section fellows introduce the design and implementation of
Configure Update, Instruction Assignment and Capability and Status
Feedback between Controller and MA.

4.1. Control Protocol Element

The Measurement Agents (MAs): implemented in network to perform measurement tasks.

-Controller: responsible for creating and assigning the measurement tasks.

4.2. Control Protocol Function

The Control Protocol is in charge for delivers Instruction Messages from Controller to MA. In this case, the Controller in response for initiate the session. In this draft, it use Controller as the https client to initiate the conversation to MA which is the server. This design brings follow advantages: Firstly, it's more effectiveness that Controller initiate the session at any time as it won't need a pool of MAs to check on Controller for new message. Secondly, It's much simple when new MA added as it only need to configure the Controller instead of start a pre-configuration session on MA.

4.2.1. Controller Behavior

Controller is responsible for creating and assigning the measurement tasks. Specifically, there are two main Controller behaviors: Firstly, the Controller initiate the session and assign the instruction to MA. Secondly, as new task is assign, the Controller update the configuration to MA.

In the protocol implementation, it's recommended to setup a state machine in the Controller to handle the communication and failure.

4.2.2. MA Behavior

The Measurement Agents (MAs) is responsible for implemented in network to perform measurement tasks. As new instruction receive from controller, MA will send a feedback message to controller.

4.3. Control Protocol Process

4.3.1. Configuration Update from Controller to MA

Configuration allows the Controller to update the MA about some or all of the information that it obtained during the bootstrapping process: the MA-ID, the (optional) Group-ID and the Control Channel. The configuration update REST-style protocol is defined to convey the

   configuration information from the Controller to the MA, via a HTTP
   PUT, as described below.

   Update the configuration from controller to MA

     PUT /ma/config/   HTTP/1.1

     Host: example.com

     Content-Type: application/yang.data+json

4.3.1.1. Request parameters

   The following are JSON parameters defined for the configuration
   update using YANG model Tree is show as follow:

   module: ietf-lmap-control

   +--rw ma-config

     +--rw ma-agent-id

     +--rw ma-control-tasks* [ma-task-name]

       +--rw ma-task-name

       +--rw ma-task-registry-entry

     +--rw ma-control-channels* [ma-channel-name]

       +--rw ma-channel-name

       +--rw ma-channel-target

       +--rw ma-channel-credientials

     +--rw ma-control-schedules* [ma-schedule-name]

       +--rw ma-schedule-name

       +--rw ma-schedule-tasks* [ma-schedule-task-name]

         +--rw ma-schedule-task-name

         +--rw ma-schedule-channels* [ma-schedule-task-source-channel-
   names]

```
         +--rw ma-schedule-task-source-channel-names

         +--rw ma-schedule-channel-interface-selection

       +--rw ma-schedule-timing* [ma-timing-name]

         +--rw ma-timing-name

         +--rw ma-timing-calendar

         +--rw ma-timing-random-spread

   +--rw ma-credentials
```

   o ma-agent-id: the assigned ID of the MA.

   o ma-control-tasks: the configuration of the control channel to be
used between the Controller and the MA, which includes the following
parameters:

   o ma-task-name: the string name of the task

   o uri: the uri of the task

   o ma-channel-name: the string name of the channel.

   o ma-channel-target: the URL of the Controller to be contacted by
the MA.

   o ma-channel-credientials: the channel credentials object.

   o ma-control-schedules: the MA schedule object.

   o ma-schedule-name: the string name of the schedule.

   o ma-schedule-tasks: the MA schedule task object.

   o ma-schedule-task-name: the string name of the MA schedule task.

   o ma-schedule-task-datasets: the MA shedule task object.

   o ma-schedule-task-channel-names: the string name of task channel.

   o ma-schedule-timing: the MA timing object.

   o ma-timing-name: the string name of the MA timing.

      o ma-timing-calendar: the MA timing calendar obejct.

      o ma-calendar-minutes: minutes number.

      o ma-calendar-seconds: seconds number.

      o ma-timing-random-spread: the random spread of the MA timing.

      o ma-credentials: the MA credentials object.

4.3.1.2. YANG Model Schema

   The YANG model Schema is show as below:

   module ietf-lmap-control {

       namespace "urn:ietf:params:xml:ns:yang:ietf-lmap-control";

       prefix "lmapc";

       import ietf-yang-types {

         prefix yang;

       }

       import ietf-inet-types {

         prefix inet;

       }

       import ietf-interfaces {

         prefix if;

       }

       contact

         "WG Web: <http://tools.ietf.org/wg/lmap/>

          WG List: <mailto:lmap@ietf.org>

          ";

       description

```
       "This module defines a data model for controlling measurement

        agents that are part of a Large-Scale Measurement Platform

        (LMAP).";

     revision "2015-05-04" {

       description

        "Initial version";

       reference

        "RFC 6020";

     }


   container ma-config {

    leaf ma-agent-id  {

       type yang:uuid;

    }

    list ma-control-tasks {

      key ma-task-name

      leaf ma-task-name {

        type string;

      }

      leaf ma-task-registry-entry {

        type  inet:uri;

      }

    }

    list ma-control-channels {
```

```
      key ma-channel-name

      leaf ma-channel-name {

        type string;

        }

      leaf ma-channel-target {

        type string ;

      }

      leaf ma-channel-credientials {

        type string

      }

    }

    list ma-control-schedules {

      key ma-schedule-name;

      leaf ma-schedule-name {

        type string;

      }

      list  ma-schedule-tasks {

        key ma-schedule-task-name;

        leaf ma-schedule-task-name {

          type string;

        }

        list ma-schedule-tasks {

          key ma-schedule-task-name;

          leaf ma-schedule-task-name {
```

```
                type string;

              }

          list ma-schedule-channels {

            key ma-schedule-task-source-channel-name

            leaf  ma-schedule-task-source-channel-names {

              type string;

            }

            leaf ma-schedule-channel-interface-selection {

              type string;

            }

          }

          list ma-schedule-timing {

            key ma-timing-name;

            leaf ma-timing-name {

              type string;

            }

            leaf ma-timing-calendar {

              type string ;

            }

            leaf ma-time-random-spread {

              type int32;

            }

          }

        }
```

```
         }

         }

      leaf ma-credentials {

        type string;

      }

    }
```

4.3.1.3. Response Codes

   Normal Response Codes: 204

   Error Response Codes: 404

4.3.1.4. JSON Content

   The following is an example of JSON format for an HTTP PUT request
   for MA 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-name": "Controller status and capabilities",

               "ma-task-registry-entry":
   "urn:ietf:lmap:control:http_Controller_status_and_capabilities"
```

```
           },
           {

             "ma-task-name": "Controller instruction",

             "ma-task-registry-entry":
     "urn:ietf:lmap:control:http_Controller_instruction"

           }

         ]

         "ma-control-channels": [

           {

             "ma-channel-name": "Controller instruction",

             "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-channels": [

                   {

                     "ma-schedule-channel-interface-selection": [1],
```

```
                    "ma-schedule-task-source-channel-names":
   ["Controller channel"]

                  }

                ]

              },

              {

                "ma-schedule-task-name": "Controller status and
   capabilities",

                "ma-schedule-channels": [

                  {


                    "ma-schedule-channel-interface-selection": [1],

                    "ma-schedule-task-source-channel-names":
   ["Controller channel"]

                  }

                ]

              },

              {

                "ma-schedule-task-name": "Controller instruction",

                "ma-schedule-channels": [

                  {

                    "ma-schedule-channel-interface-selection": [1],

                    "ma-schedule-task-source-channel-names":
   ["Controller channel"]

                  }

                ]
```

```
                  }

              ]

              "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": { }

        }

      }
```

4.3.2. Instruction Assignment from Controller to MA

   The Instruction is the description of the Measurement Tasks for a
   Measurement Agent to do and the details of the Measurement Reports
   for it to send, which is realized by a HTTP POST transaction.

   Send measurement instruction from controller to MA

   POST /ma/ins/ HTTP/1.1

   Host: example.com

   Content-Type: application/yang.data+json

4.3.2.1. Request parameters

   module:ietf-lmap-control-assign

```
   +--rw ma-task

      +--rw ma-task-name

      +--rw ma-task-registry

      +--rw ma-task-operation

      +--rw ma-task-cycle-id

      +--rw ma-schedule

         +--rw ma-schedule-name

         +--rw ma-schedule-task* [ma-task-name]

            +--rw ma-task-name

            +--rw ma-task-registry

            +--rw ma-task-options

            +--rw ma-task-cycle-id

         +--rw ma-schedule-timing

            +--rw ma-timer-periodic

            +--rw ma-timer-randomness

      +--rw ma-channel

         +--rw ma-channel-name

         +--rw ma-channel-target

         +--rw ma-channel-certificate

         +--rw ma-channel-interface-name

         +--rw ma-channel-connect-always

      +--rw ma-supression

         +--rw ma-suppression-enabled

         +--rw ma-suppression-start
```

```
+--rw ma-suppression-end

+--tw ma-suppression-task-names

+--rw ma-suppression-schedule-names
```

ma-tasks: the list of measurement tasks contained in the nstruction, where each measurement task contains the following four parameters:

ma-task-name: the string name for a specific easurement task, to be used as reference in other places, such as easurement schedules and suppressions.

ma-task-registry: the URN Uniform Resource Name) of the measurement task as defined by [I-D.draft-ippm-registry].

ma-task-options: the list of input arameters that are to be used when performing the measurement task, hich are specific to each measurement task in question and defined y [I-D.draft-ippm-registry].

ma-task-cycle-id(optional): the dentification ID for a group of measurement tasks with comparable ptions, which is manually incremented when an Option change is mplemented which could mean that two sets of results should not be irectly compared.

ma-schedules: the list of measurement schedules n the instruction, where each measurement schedule contains the ollowing parameters:

ma-schedule-name: the string name for a pecific measurement schedule, to be used as reference in other laces, such as suppressions.

ma-schedule-tasks: the list of measurement tasks to be performed for this schedule, where each scheduled measurement task contains the following parameter in addition to its name, options and cycle-id:

ma-schedule-task-reports: the report channel settings for the measurement reporting, which in turn contains an optional filter configuration for a specified group (default is all) of measurement output to a named eport channel.

ma-channel: the report channel to be used by the easurement schedules to configure the measurement reporting for the instruction,

which is also used to define the control channel by configuration request from Controller to the MA.

ma-channel-name: the sting name of the channel.

ma-channel-target: the URL of the Controller to be contacted by the MA.

ma-channel-timing: the timing arrangements indicating when to contact the Controller by the MA, whose in turn includes the following parameters:

ma-timing- name(optional): the string name of the time setting.

ma-timing- arrangement: the arrangements for the timing of Controller communications, whose value can be one of the following parameters:

ma-timing-periodic: the periodic timing arrangement for the Controller communications, which in turn includes the following parameters:

ma-periodic-start(optional): the time of the day (in milliseconds) when the communication starts. If absent, the period starts immediately.

ma-periodic-end(optional): the time of day (in milliseconds) when the communication ends. If absent, the communication is not stopped unless the MA is otherwise instructed.

ma-periodic-interval: the interval time (in milliseconds) of periodic communications.

ma-timing-calendar: the communication happens according to a calendar-like timing arrangements.

ma-timing-one-off: the communication happens once and for all.

ma-timing-immediate: the communication happens immediately.

ma-timing-randomness: the randomness configuration to be added onto each communication event.

ma-channel-interface-name: the name of MA's interface to be used when contacting the Controller.

   ma-channel-connect-always: the indicator whether or not to keep
   the connection to the Controller. If absent, the connection is
   reopened whenever new communication is pending and closed when it is
   over.

   ma-suppression: the measurement tasks/schedules to be suppressed
   by the instruction. In addition to the lists of measurement
   tasks/schedules, it also contains the following parameters:

   ma-suppression-enabled(optional): The bool indicator of whether or
   not to enable the local suppression by the MA. For example, a later
   unsuppression to an earlier suppression instruction can be achieved
   by setting this indicator to 'false'. Default is false.

   ma- suppression-start(optional): The time when to start the
   suppression action. Default is immediate.

   ma-suppression-end(optional): The time when to end the suppression
   action. Default is infinite.

4.3.2.2. YANG Model Schema

```
container ma-task {

  leaf ma-task-name {

    type  string;

  }

  leaf ma-task-registry{

    type  string;

  }

  leaf ma-task-operation{

    type  string;

  }

  ma-task-cycle-id {

    type  string;

  }
```

```
list ma-schedule {

  key ma-schedule-name

  leaf ma-schedule-name {

     type string;

  }

  list ma-schedule-task {

    key ma-task-name;

    leaf ma-task-name{

       type string;

    }

    leaf ma-task-registry  {

       type inet:uri;

    }

    leaf ma-task-options  {

      type string;

    }

    leaf ma-task-cycle-id {

      type string;

    }

    container  schedule-timing {

     leaf ma-timer-periodic {

        type int32;

      }

     leaf ma-timer-randomness {
```

```
            type int32;

          }

        }

      }

     container ma-channel {

        leaf  ma-channel-name {

           type string;

        }

        leaf ma-channel-target {

           type string;

        }

        leaf ma-channel-certificate {

          type string;

        }

        leaf  ma-channel-interface-name   {

          type string;

        }

        leaf  ma-channel-connect-always {

          type boolean;

        }

      }

     container ma-supression {

        leaf   ma-suppression-enabled {

          type boolean;
```

```
          }

          leaf  ma-suppression-start {

            type yang:date-and-time;

          }

          leaf ma-suppression-end {

            type yang:date-and-time;

          }

          list ma-suppression-task-names {

           key name;

           leaf name {

              type string;

           }

           }

          list ma-suppression-schedule-names {

           key name;

           leaf name {

             type string;

           }

          }

        }

      }

      }

    }
```

4.3.2.3. Response codes

      Normal Response Codes: 201

       Error Response Codes: 400

4.3.2.4. JSON Content

```
    {

        "ma-task": {

            "ma-task-name":"",

            "ma-task-registry":"",

            "ma-task-options":"",

            "ma-task-cycle-id":""

        }, "ma-schedule:{

            "ma-schedule-name":"",

            "ma-schedule-tasks":[{

                "ma-task-name":"",

                "ma-task-registry":"",

                "ma-task-options":"",

                "ma-task-cycle-id":""

            },

            {

                "ma-task-name":"",

                "ma-task-registry":"",

                "ma-task-options":"",

                "ma-task-cycle-id":""

            },
```

```
                     ...
                ],

                "ma-schedule-timing":{

                     "ma-timing-periodic":"",

                     "ma-timing-randomness":""

                }

          },

           "ma-channel":{

                "ma-channel-name":"",

                "ma-channel-target":"",

                "ma-channel-certificate":"",

                "ma-channel-timing":"",

                "ma-channel-interface-name":"",

                "ma-channel-connect-always":""

           },

           "ma-suppression":{

      "ma-suppression-enabled":"",

      "ma-suppression-start":"",

      "ma-suppression-end":"",

      "ma-suppression-task-names":[

                     { "task-name":""},

                     { "task-name":""},

                     ...

      ],
```

```
        "ma-suppression-schedule-names":[

                    { "schedule-name":""},

                    { "schedule-name":""},

                    ...

        ]     }

        }
```

4.3.3. Feedback from MA to Controller

   The Control Protocol enable Capabilities Failures and Status
   information feedback from MA to Controller whenever requested.

   For the failures information, it must consider two typical failure
   scenes:

   1. Communication failure: the communication with the Controller, MP.

   2. Capability failure: disability for implementing the measurement
   task etc.

   Each failure has a unique code. For the status information, the MA
   will send its current status or some own events log information to
   the Controller. That status information like 'keep alive' can be
   defined in CP for ensuring to notify the Controller that MA is in
   good condition now. The event log information can include such events
   (not limited): beginning measurement, finishing measurement, recovery
   from failure.

   Capability and status feedback is returned to the Controller from the
   MA whenever requested, which is implemented by an HTTP Get
   transaction.

4.3.3.1. Request

      GET /ma/capabilities HTTP/1.1

      Host: example.com

       Accept: application/yang.data+json

Get failure information API: GET /ma/failure

Get logging information API: GET /ma/logging

4.3.3.2. Feedback Parameters

```
 modules: ietf-lmap-state

 +--ro ma-status-and-capabilities

    +--ro ma-agent-id

    +--ro ma-device-id

    +--ro ma-hardware

    +--ro ma-firmware

    +--ro ma-version

    +--ro ma-interfaces* [ma-interface-name]

      +--ro ma-interface-name

      +--ro ma-interface-type

    +--ro ma-last-measurement

    +--ro ma-last-report

    +--ro ma-last-instruction

    +--ro ma-last-configuration

    +--ro ma-supported-tasks* [ma-task-name]

      +--ro ma-task-name

      +--ro ma-task-registry
```

4.3.3.3. Response Code

Code 200 for success.

Code 400 for failure.

4.3.3.4. Response YANG Model

```
   YANG Model

   module ietf-lmap-state {

       namespace "urn:ietf:params:xml:ns:yang:ietf-lmap-state";

       prefix "lmapc";

       import ietf-yang-types {

         prefix yang;

       }

       import ietf-inet-types {

         prefix inet;

       }

       import ietf-interfaces {

         prefix if;

       }

       contact

         "WG Web:   <http://tools.ietf.org/wg/lmap/>

          WG List:  <mailto:lmap@ietf.org>

          ";

       description

         "This module defines a data model for controlling measurement

          agents that are part of a Large-Scale Measurement Platform

          (LMAP).";

       revision "2015-05-04" {

         description
```

```
          "Initial version";
        reference
         "RFC 6020";
      }
      container  ma-status-and-capabilities {
        leaf ma-agent-id {
          type yang:uuid;
        }
        leaf ma-DEVICE-id {
          type inet:uri;
        }
        leaf ma-hardware {
          type string;
        }
        leaf ma-firmware {
          type string;
        }
        leaf ma-version  {
           type string;
        }
        list ma-interfaces {
           key ma-interface-name;
           leaf ma-interface-name {
             type string;
```

```
            }
            leaf ma-interface-type {
              type string;
            }
         }
        leaf ma-last-measurement {
           type string;
        }
        leaf ma-last-report {
          type string;
        }
        leaf ma-last-instruction {
          type string;
        }
        leaf ma-last-configuration {
          type string;
        }
        list ma-suported-tasks {
           key ma-task-name;
           leaf ma-task-name {
             type string;
           }
           leaf ma-task-registry {
             type inet:uri;
```

```
            }
          }
        }
      }
```

4.3.3.5. JSON Content

```
      {
          "ma-status-and-capabilities": {
              "ma-agent-id": "",
              "ma-device-id": "",
              "ma-hardware": "",
              "ma-firmware": "",
              "ma-version": "",
              "ma-interfaces": [
                  {
                      "ma-interface-name": "",
                      "ma-interface-type": ""
                  }
              ],
              "ma-last-measurement": "",
              "ma-last-report": "",
              "ma-last-instruction": "",
              "ma-last-configuration": "",
              "ma-supported-tasks": [
                  {
```

```
                "ma-task-name": "",

                "ma-task-registry": ""

            },

            {

                "ma-task-name": "",

                "ma-task-registry": ""

            },

            {

                "ma-task-name": "",

                "ma-task-registry": ""

            },

            {

                "ma-task-name": "",

                "ma-task-registry": ""

            },

            {

                "ma-task-name": "",

                "ma-task-registry": ""

            }

        ]

    }

}
```

The response of get failure information API is in JSON format:

```
{
```

```
   "failure code 1":"no spare CPU cycles",

   "failure code 2":"out of spare memory",

   "failure code 3":"Collector is not responding",

      ...

}
```

The response of get logging information API is as follows:

```
{

   "ma-log-agent-id":"",

   "ma-log-event-time": "",

   "ma-log-code":"",

   "ma-log-description":""

}
```

5. Report Protocol

   This section discuss design and implementation of Report Protocol
   based on REST-style. The Report Protocol is being used between
   Collector and MA to report Measurement Results.

   The section fellows introduce the Format and Code of the Report
   Protocol.

     5.1. Report Protocol Element

   The Measurement Agents (MAs): implemented in network to perform
   measurement tasks.

   Collector: in charge of collecting and storing measurement results.

5.2. Report Protocol Function

Report Protocol is response for delivering Report(s) from MA to
Collector which runs over the Report Channel. In Report Protocol, the
MA as the Http client to initiate session to Collector which as the
server.

5.2.1. Collector Behavior

The Collector accepts measurement task report from MA.

5.2.2. MA Behavior

The MA report measurement task report to collector.

5.3. Report Protocol Process

5.3.1. Report from MA to Controller

5.3.1.1. Report Request

Request API:

POST /collector/report/ HTTP/1.1

Host: example.com

Content-Type: application/yang.operation+json

5.3.1.2. Report parameters using YANG tree

```
 modules: ietf-lmap-report

+--ro ma-report

  +--ro ma-report-date

  +--ro ma-report-agent-id

  +--ro ma-report-group-id

  +--ro ma-report-tasks* [ma-task-name]

    +--ro ma-task-name

    +--ro ma-task-registry
```

```
      +--ro ma-task-options* [name]

         +--ro name

         +--ro value

    +--ro ma-task-suppress-by-default

    +--ro ma-task-cycle-id

    +--ro ma-report-task-column-labels* [ma-task-name]

      +--ro ma-task-name

      +--ro ma-report-column-label* [name]

         +--name

    +--ro ma-report-task-rows* [ma-task-name]

      +--ro ma-task-name

      +--ro ma-report-result-time

      +--ro ma-report-conflicting-tasks

      +--ro ma-report-result-cross-traffic

      +--ro ma-report-result-values
```

5.3.1.3. YANG Model Schema

```
   module ietf-lmap-report {

       namespace "urn:ietf:params:xml:ns:yang:ietf-lmap-report";

       prefix "lmapc";

       import ietf-yang-types {

         prefix yang;

       }

       import ietf-inet-types {

         prefix inet;
```

```
      }

      import ietf-interfaces {

        prefix if;

      }

      contact

        "WG Web:    <http://tools.ietf.org/wg/lmap/>

         WG List:  <mailto:lmap@ietf.org>

         ";

      description

        "This module defines a data model for controlling measurement

         agents that are part of a Large-Scale Measurement Platform

         (LMAP).";

      revision "2015-05-04" {

        description

         "Initial version";

        reference

         "RFC 6020";

      }

      container ma-report {

      leaf ma-report-date {

        type yang:date-and-time;

      }

      leaf ma-report-agent-id {

         type yang:uuid;
```

```
          }

          leaf ma-report-group-id {

             type string;

          }

          list ma-report-tasks {

             key ma-task-name;

             leaf ma-task-name {

                type string;

             }

             leaf ma-task-registry {

                type inet:uri;

             }

             list ma-task-options  {

              key name;

              leaf name {

                 type string;

              }

              leaf value {

                 type string ;

              }

             }

             leaf ma-task-suppress-by-default {

               type boolean;

             }
```

```
          leaf ma-task-cycle-id  {

            type string;

          }

          list ma-report-task-column-labels {

             key ma-task-name

             leaf ma-task-name {

               type string;

             }

             list ma-report-column-label {

                key name;

                leaf name {

                 type string;

                }

             }

          }

          list ma-report-task-rows  {

             key ma-task-name

             leaf ma-task-name {

               type string;

             }

             leaf ma-report-report-time {

                type string;

             }

             leaf ma-report-conflicting-tasks {
```

```
          type string;

        }

        leaf ma-report-result-cross-traffic {

          type string;

        }

        leaf ma-report-result-values {

           type string;

        }

      }

    }


  }

}
```

5.3.1.4. Response Code

      Code 200 for success.

      Code 400 for failure.

5.3.1.5. Response JSON content:

```
      {

          "ma-report-date": "",

          "ma-report-agent-id": "",

          "ma-report-group-id": "",

          "ma-report-tasks": [

              {
```

```
                "ma-report-task-config": {

                    "ma-task-name": "",

                    "ma-task-registry": "",

                    "ma-task-options": [

                        {

                            "name": "",

                            "value": ""

                        },

                        {

                            "name": "",

                            "value": ""

                        }

                    ]

                }

            }

        ],

        "ma-task-suppress-by-default": "",

        "ma-task-cycle-id": "",

        "ma-report-task-column-labels": [ ],

        "ma-report-task-rows": [

            {

                "ma-report-result-time": "",

                "ma-report-conflicting-tasks": "",

                "ma-report-result-cross-traffic": "",
```

```
                "ma-report-result-values": ""

           }

       ]

   }
```

6. Security Considerations

   The REST API could be protected by several ways. One common solution
   is to provide authentication mechanism for all the HTTP request.
   Encryption mechanism could also be provided by HTTPS, the credential
   can be employed for authentication.

7. IANA Considerations

   There is no IANA action in this document.

8. Conclusions

   This draft introduced a protocol achieving LMAP along with
   implementation using RESTconf and YANG model.

9. References

   9.1. Normative References

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

   [2]   Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax
         Specifications: ABNF", RFC 2234, Internet Mail Consortium and
         Demon Internet Ltd., November 1997.

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

   [RFC2234] Crocker, D. and Overell, P.(Editors), "Augmented BNF for
             Syntax Specifications: ABNF", RFC 2234, Internet Mail
             Consortium and Demon Internet Ltd., November 1997.

   [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the
             Network Configuration Protocol (NETCONF)", RFC 6020,
             October 2010.

[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-11 (work in progress), Feburary 2015.

[I-D.ietf-lmap-information-model] Burbridge, T., Eardley, P.,Bagnulo,
          M., and J. Schoenwaelder, "Information Model for Large-
          Scale Measurement Platforms (LMAP)", draft-ietf-lmap-
          information-model-03 (work in progress), January 2015.

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

[RFC7213] R. Fielding, J. Reschke "Hypertext Transfer
          Protocol(HTTP/1.1): Semantics and Content"

[RFC7159] T. Bray, "The JavaScript Object Notation (JSON) Data
          Interchange Format"

[draft-starkcarey-lmap-protocol-criteria-01] B. Stark, T. Carey "LMAP
Protocol Selection Criteria"

election Criteria"

10. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

11. Appendix A. Reply to Protocol Criteria

Control Protocol Criteria

Mandatory Criteria

[CP-MUST-1]-Yes, the Controller as HTTP client to initiate the
session to MA which as HTTP server.

[CP-MUST-2]-Yes, MA can initiate Feedback to Controller.

[CP-MUST-3]-Yes, credential employed for authentication by HTTPS.

[CP-MUST-4]-Yes by YANG.

Comparative Criteria

[CP-DIFF-1] 1

[CP-DIFF-2] 1

[CP-DIFF-3] Yes

[CP-DIFF-4] JSON and REST over HTTPs

[CP-DIFF-5] Header contain about 100bytes(host\content type\accept)
[CP-DIFF-6] Header contain about 100bytes(host\content type\accept)
[CP-DIFF-7] Restful widely used. Meantime, Chinacache has been used
it as a trial.

[CP-DIFF-8]Test tools for JSON. More is under developed.

[CP-DIFF-9] yes on Github

[CP-DIFF-10] JSON and HTTPs, also we developed open source program.

[CP-DIFF-11] yes

[CP-DIFF-12] yes

[CP-DIFF-13] JSON

Report Protocol Criteria

Mandatory Criteria

[RP-MUST-1] Yes, the MA as HTTP client to initiate the session to
Collector which as HTTP server.

[RP-MUST-2] JSON and RESTful over HTTPs

[RP-MUST-3] Yes by YANG

Comparative Criteria

[RP-DIFF-1] TCP

[RP-DIFF-2] Yes

[RP-DIFF-3] 1

[RP-DIFF-4] Yes

[RP-DIFF-5] Yes

[RP-DIFF-6] Yes

[RP-DIFF-7] about 100 bytes

[RP-DIFF-8] REST is used widely

[RP-DIFF-9] Test tools for JSON, certification program [RP-DIFF-10] Yes [RP-DIFF-11] JSON and HTTPs, also we developed open source program.

[RP-DIFF-12] JSON

Authors' Addresses

   Vic Liu
   China Mobile
   32 Xuanwumen West AVE, Xicheng, Beijing

   Email: liuzhiheng@chinamobile.com

   Lingli Deng
   China Mobile
   32 Xuanwumen West AVE, Xicheng, Beijing

   Email: denglingli@chinamobile.com

   Shu Liu
   China Academy of Information and Communication Technology

   Email: liushu@catr.cn

   Dapeng Liu

   Email: max@dotalks.com

Cathy Li
China Cache

Email: cathy.li@chinacache.com

                        Internet Measurement System
                draft-ooki-lmap-internet-measurement-system-03

Abstract

   This document describes an experience of Japanese Internet
   measurement system to measure end-to-end performance of user's
   experience. We have developed the system toward the enhancement of
   the network performance in an ISP since October 2013. The systems
   and the considerations about the Internet measurement are introduced
   along with our current status. This document is expected to be
   useful for the standardization of Internet measurements.

Status of This Memo

Copyright Notice

the Trust Legal Provisions and are provided without warranty as
described in the Simplified BSD License.

Table of Contents

1. Introduction

   In Japan, it is common to use a high speed Internet such as 100Mbps
   and 1Gbps as an ISP's customer connection. Users only know the
   maximum bandwidth of the last one mile for the ISP connection. The
   maximum bandwidth value is ranging from 100Mbps to 2Gbps in ISP's
   price plan as a FTTH connection.

   Of course the end-to-end performance of actual Internet connection is
   below the bandwidth value. Internet users can obtain actual
   performance depends on various ISP conditions such as congestions.
   Internet users don't know the performance of the actual network.

On the other hand, ISPs also don't know the quality that Internet
users experience. For the ISP's point of view, it is important to
understand the service quality for its customers in order to design
its network properly. For this reason, it is necessary to measure
the actual performance of typical Internet users.

The Large-Scale Measurement of Broadband Performance (LMAP) working
group is formed to standardize a large scale measurement system to
measure broadband network performance. The current LMAP WG focus on
the information model, data model language, the protocols in a
certain ISP network. However, the LMAP WG does not focus on the
measurement of the global end-to-end performance at the moment. We
believe that either way someday it will be necessary to establish a
method for the Internet measurement and the standardization of the
end-to-end performance measurement, that is not closed to a certain
ISP.

This document describes the Internet measurement system and our
considerations for the end-to-end measurement. Our measurement
requirements can be useful for LMAP framework. We have measured the
end-to-end performance by using Internet measurement system we have
been operating since 2011. We expect the experience of our case can
contribute to the standardizations in LMAP WG and the enhancement of
network operation from ISP's perspective.

2. Motivation of Internet Measurement

The LMAP WG describes some use cases for the Large-scale Measurement
of Broadband Performance [RFC7536][I-D.deng-lmap-collaboration].
There are three reasons that we, ISPs, need to measure the end-to-end
performance of user experience of its access services.

First, ISPs want to keep the customer satisfaction. Typically ISPs
provide the list of maximum bandwidth and the service prices, such as
the estimated total fee and the discount rate after the result of a
cash back campaign. Japanese users select a ISP based on only those
information without knowing the end-to-end performance results. The
poor performance causes the lower customer satisfaction.

On the other hand, the Ministry of Internal Affairs and
Communications in Japan discuss the ideal measurement methods of the
end-to-end performance about the mobile network in the research
society. The organization has yet to be discussed fixed network.
The researchers are planning to measure the mobile network
performance on the 1,500 measurement points in the main Japanese
area. They use the tool Federal Communications Commission developed
for measurement of the end-to-end performance. The mobile network

operator in Japan may be required to publish the actual performance
in addition to the best effort performance.

Second, contents providers are beyond the control of ISPs. The
traffic volume of Contents Delivery Network (CDN) providers such as
AKAMAI and LEVEL3 is increasing in the Internet in recent years. How
much users are connecting to which contents providers impact the end-
to-end performance. ISPs need to understand their behavior to decide
ISP's strategies and operation.

Third, we would like to support the public evaluations of ISPs. Some
contents service provider e.g., Google or Sandvine[google][sandvine],
presented the reports about Internet traffic and ISP performance
based on each criterion recently. The Google report presents the
results of multiple ISPs measured in for each locations in USA.
People in the world can browse the reports on the Internet. These
reports will have huge impact on user's choice of the ISP selection.
We would like to double-check by using our performance data in order
to confirm whether the reports can be reliable or not. If we can
find the difference of the performance results between our data and
the reports, we might be able to review whether our measurement
methods are mistaken or not. It is also better for an ISP to
investigate and comprehend the status of end-to-end performance
between ISPs. So, we have to measure the end-to-end performance by
ourself.

Hence, ISPs should measure the end-to-end performances from end users
to multiple content providers accurately while comparing with other
ISPs' performance. Then, ISPs can show a performance of the actual
network to build brand value compared with other providers.

3. Framework of Internet Measurement System

We introduce the framework of Internet Measurement System in this
section. The words, such as Measurement Agent, Controller, and
Collector conforms to the glossary of the LMAP document[I-D.ietf-
lmap-framework].

3.1. Measurement Agent

The MA has the functions that receive instructions from Controller
Server (described below), performs measurement tasks, and sends the
measurement results to the Collector Server.

3.1.1. Specification of the MA

   We used a Japanese product, called OpenBlocks [plathome], which is
   the Linux box with Dual Core Marvell ARMADA XP 1.33GHz, 1GB SDRAM
   memory. We selected the box as the MA because of the affordable
   price, software stability, small form factor, flexible functionally,
   and extendability. The MA needs some CPU power in order to connect
   PPPoE access line and download tens of contents on the Internet. The
   OpenBlocks stacks CPU enough to archive them.

3.1.2. Configuration

   We introduce information configured on the MA in this section.

   o MA's ID

   We have to setup the MA's ID. The ID has to be a unique among MAs in
   order for Controller server to distinguish MAs. The information is
   described in the "/etc/hostname" on Linux File System. The naming
   rule is based on the location of MA, the types of line, and the plan,
   etc.

   On the other hand, MA doesn't have a group ID. The ID in our
   measurement system is under only Controller Server.

   o HTTP Get Tasks

   MA automatically gets the measurement tasks from Controller Server
   every five minutes. MA sends the request about the tasks and
   schedule to Controller Server by HTTP. Controller Server returns the
   tasks decided based on SCHEDULE Table to the MA.

   o Convert measurement results

   MA automatically converts from raw data to the type of JSON data.
   JSON is a lightweight data-interchange format. We selected it as our
   measurement data model language because of readability, simple
   format, and easy data cleaning to analyze the measurement data. On
   the other hand, the LMAP WG selected YANG Data Model [I-D.ietf-lmap-
   yang]. We need to consider which we should use YANG data model or
   not. The convert process is executed every one minute. The
   converted JSON data is written in the measurement result files. An
   example of JSON data type of the results of wget raw data is as
   follows.

   {"host":"tokyo-xXx01","filename":"tokyo-xXx01_ISP_target_wget_2014120
   3235011.log.ok","result":"ok","line":"2","message":"2014-12-03
   23:50:18 (10.1 MB/s) - '/dev/null' saved [67206439/67206439]"}

o Data Collector and Submitting

MA automatically and efficiently collects and submits the measurement
data. To make it realized, we selected the fluentd which is an open
source data collector, The software lets Collector Server unify the
data collection and understanding of data. The reason we selected it
is that the software is reliable, stable, and simple of
implementation enough to control hundreds of MAs. The data
collecting tool of many products and system in Japan is implemented
by fluentd. The above measurement result files are submitted to
Collector Server by the process as soon as the files are created. An
example of the running flutend process is as follows.

```
/usr/bin/ruby1.9.1 /usr/local/bin/fluentd --daemon /var/lib/fluent/
fluentd.pid --user fluent --group fluent --config /etc/fluent/
fluent.conf --log /var/log/fluent/fluent.log -vv
```

o Self Check

The MAs check whether the above processes for measurement service are
down or not on the regular interval. If a process is down, the MA
transmits the message about the message logs to Controller Server.

3.1.3. Location of the MA

We have distributed MAs on many places all over Japan. The number of
locations is approximately 150 in June 2015. The number of our MAs
will be increasing in approximately 200 by the end of this year. MAs
are located in houses where the residents can respond our requests
(e.g., not turning off the power to constantly perform the
measurement) to manipulate the device.

3.2. Controller Server

The Controller Server is a Linux server. The Controller Server has
the functions to instruct the MAs and receives the HTTP GET requests
from MAs. The Controller Server has 3 tables of database implemented
by MySQL to instruct MAs.

3.2.1. Control of MAs

The Controller Server manage MAs by using two tables (MA Table and
GROUP Table). A example of the MA Table is as follows. MA_ID is a
key identifying MA. TYPE is a kind of network type. MODE expresses
the type of the measurement.IF MODE is 0, it means MA is the
measurement mode. The mode is the status that MAs are performing a
measurement task. If MODE is 1, it means MA is the maintenance mode.
The mode is the status that MA stop performing a measurement task.

In case of MODE 1, the MA automatically connect to Controller Server
by using ssh protocol. We can login to the MA of MODE 1 from
Controller Server and change the configuration. We can manage the
behavior for MAs by switching the MODE.

MA_ID	TYPE	AREA	OS_TYPE	MODE	GET_SCHEDULE_TIME
tokyo-nFh04	flets	tokyo	Debian 7	1	2014-12-08 23:21:00
osaka-nFs01	flets	osaka	Debian 7	1	2014-12-08 23:22:00

Table 1: MA Table

A example of the GROUP Table is as follows. The GROUP_ID is a key
record grouping MAs. MAs are sure to belong one group at least. MA,
tokyo-nFh04, belongs to the group-id1. The GROUP_INFO is the
remarks. We can set the information of the group which MAs belong in
the column

GROUP_ID	MA_ID	GROUP_INFO
group-id1	tokyo-nFh04	Group 01
group-id2	osaka-nFs01	Group 02

Table 2: GROUP Table

3.2.2. Control of the Assigned ISP

We set the information of ISP accounts to assign MAs on the
Controller Server. MAs automatically download the information from
Controller Server. Of course, assigned ISP account is unique of all
ISP accounts for the measurements. The table includes the column of
multiple assigned information so that the duplicate use does not
happen. A example of the ISP Table is as follows. ASSIGN_ID is a
key record within the table. MEASURE_ISP is the service name of ISP.
ISP_ID is the unique ID to connect to the ISP network. PASSWORD is
the password of the ISP_ID. ASSIGN_STATUS, ASSING_MA, and
ASSIGN_TIME are the assigned information at that time. If the
ASSIGN_STATUS is 1, that means a MA use the ISP_ID.

```
+-------+--------+----------+-------+----------+---------+---------+
| ASSIG | MEASUR |  ISP_ID  | PASSW | ASSIGN_S | ASSIGN_ | ASSIGN_ |
| N_ID  | E_ISP  |          |  ORD  |  TATUS   |   MA    |  TIME   |
+-------+--------+----------+-------+----------+---------+---------+
|   1   |  OCN   | abc123@oc | abc12 |    1     | tokyo-  | 2014-12 |
|       |        |  n.ne.jp  | 3def  |          |  nFh04  | -08 23: |
|       |        |          |       |          |         |  21:05  |
|   2   |  OCN   | ghi456@oc | ghi45 |    0     |         |         |
|       |        |  n.ne.jp  | 6jkl  |          |         |         |
+-------+--------+----------+-------+----------+---------+---------+
```

                       Table 3: ISP Table

3.2.3. Setting the Measurement Task and Measurement Schedule

   We set the measurement tasks to instruct MAs on the Controller
   Server. MAs automatically download the task from Controller Server
   by the fixed time. We need to set a measurement schedule with the
   measurement task at the same time. A example of the MEASUREMENT
   SCHEDULE Table is as follows. SCH_ID is a key record within the
   table. LINE_TYPE is a type of network provided by a network service.
   ISP is a Internet service provider to perform the measurement tasks.
   SCRIPT is the script file of the measurement tasks described by some
   programming languages. PARAM is a parameter file required for
   performing measurement tasks. START_TIME is the time when MAs start
   performing a measurement task. END_TIME is the time when MAs stop
   performing a measurement task.

```
+------+--------+--------+----+----------+-------+---------+--------+
| SCH_ | GROUP_ | LINE_T | IS |  SCRIPT  | PARAM | START_T | END_TI |
|  ID  |   ID   |  YPE   |  P |          |       |   IME   |   ME   |
+------+--------+--------+----+----------+-------+---------+--------+
|   1  | group- | flets  | IS | measure1 | param | 00:00:0 | 00:00: |
|      |  id1   |        | P1 |   .sh    |   1   |    0    |   00   |
|   2  | group- | flets  | IS | measure2 | param | 00:00:0 | 00:00: |
|      |  id2   |        | P2 |   .sh    |   2   |    0    |   00   |
+------+--------+--------+----+----------+-------+---------+--------+
```

              Table 4: MEASUREMENT SCHEDULE Table

3.2.4. Receiving the Requests

   On the Controller Server, a httpd program is running as a daemon that
   executes continuously in the background to handle requests. The
   Controller Server returns the appropriate measurement tasks and
   measurement schedules to MAs in response to HTTP GET requests. The
   MA which complete own measurement task receives a new measurement

task continuously. MAs can start performing the next measurement
tasks continuously.

3.3. Collector Server

The Collector Server receives the measurement results from MAs
through fluentd process. The fluentd process is running as a daemon
that executes continuously in the background to handle the
measurements data. The details of the measurement results received
by fluentd process are listed below.

20141214230628+0900 measure.tokyo-nFh04 {"host":"tokyo-
nFh04","filename":"tokyo-nFh04_ISP_DEST_wget_20141214230450.log.ok","
result":"ok","message":"2014-12-14 23:06:18 (745 KB/s) - '/dev/null'
saved [67206439/67206439]"}

20141214230902+0900 measure.tokyo-nFh04 {"host":"tokyo-
nFh04","filename":"tokyo-nFh04_ISP_DEST_wget_20141214230731.log.ok","
result":"ok","message":"2014-12-14 23:08:52 (811 KB/s) - '/dev/null'
saved [67206439/67206439]"}

A example of the directory structure of the stored measurement
results is as follows. A MA's measurement result file is created by
the day.

/data/MA's-ID/measure_result_MA's-ID.DATETIME.log

When the Collector Server receives the measurement results, the
server creates the directory of the MA-ID of MA and the measurement
result files. The measurement results are stored in the data
directory.

3.4. Architecture

The architecture of the measurement system is composed of MAs,
Controller Server, and, Collector Server.

```
     +---------------+                 +---------------+
     |   Controller  |                 |   Collector   |
     |               |                 |               |
     |     Server    |                 |     Server    |
     +---------------+                 +---------------+
            ^                                  ^
            |                                  |
            | Get Measurement        Sending   |
            | Tasks and Measurement  the Results|
            | Schedules                        |
            |           +-------------+        |
```

```
             |        | Measurement |        |
      +----------    |             | -----------+
             |        |   Agents    |
             +-------------+
                 |        ^
      Perform Only |       | Obtain
    Active Measurement |   | Results
                 v        |
   +------------------------------------------------+
   |                                                |
   |              FTTH Access Line                  |
   |                                                |
   +------------------------------------------------+
                 |        ^
                 |        |
                 |        |
                 v        |
   +------------------------------------------------+
   |                                                |
   |                ISP network                     |
   |                                                |
   +------------------------------------------------+
                 |        ^
                 |        |
                 |        |
                 v        |
   +------------------------------------------------+
   |                                                |
   |                 Internet                       |
   |                                                |
   +------------------------------------------------+
                 |        ^
                 |        |
                 |        |
                 v        |
          +----------------------+
          |   Target Contents    |
          +----------------------+
```
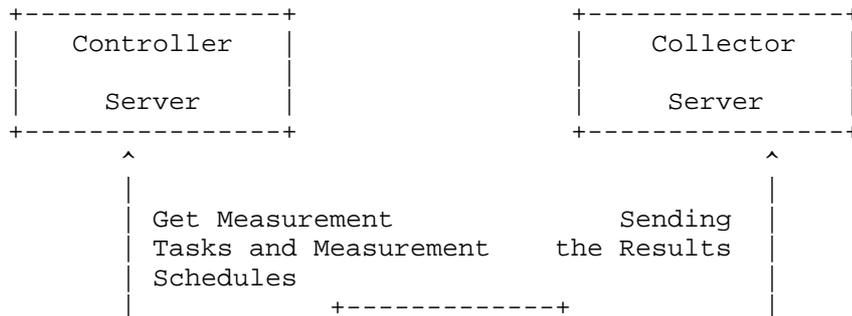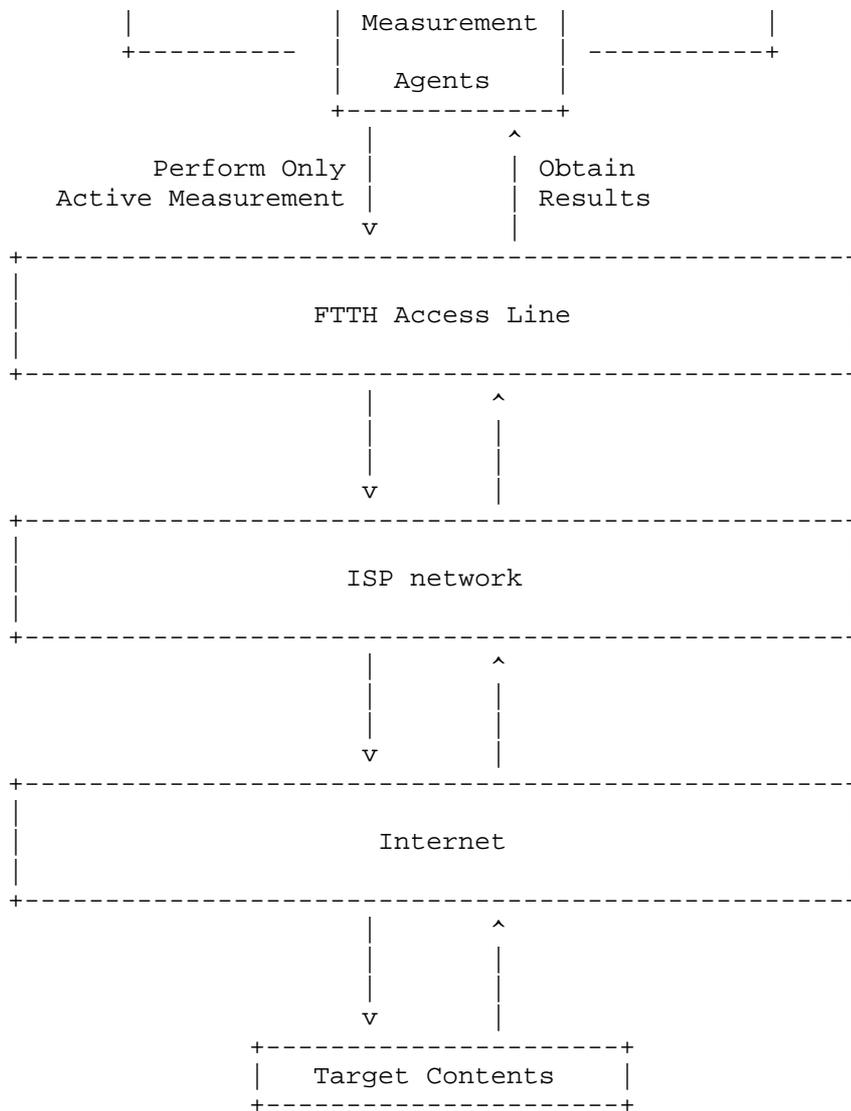
Figure 1: Architecture of the Internet Measurement System

We need to import Controller Server the record of MA's configuration
in MA Table, GROUP Table, and MEASUREMENT SCHEDULE Table before the
MA is powered on.

When a MA is powered on, it tries to establish the FTTH access PPPoE
connection with the ISP. After obtaining an IP address, it

automatically connect the Controller Server and gets the
configuration by HTTP. If the value of MODE column in MA Table is 1,
the MA automatically gets the maintenance mode. If the value of that
is 0, the MA automatically gets the measurement mode and start
downloading the measurement tasks that is configured in MEASUREMENT
SCHEDULE Table.

The MA prepares for the measurement tasks, performs the tasks for
Measurement Target Contents actively, and collects the measurement
results.

After the completion of the measurement tasks, the MA sends the
measurement results to Collector Server using fluentd process. and
submits the request for downloading the next measurement tasks to
Controller Server using HTTP.

When the specification of the LMAP WG's protocol and framework is
finished, we will deploy the protocol in our measurement system.

4. Operation of Internet Measurement System

   We introduce the operation of Internet Measurement System we have
   been operating since October 2013 in this section.

4.1. Measurement Performance Metrics

   The MAs perform only active measurements for Target Contents.
   Examples of the Target Contents include Video Streaming files, OS
   update files, and the test server for performance measurement in a
   local ISP network.

   In our measurements, the measurement performance metrics are as
   follows.

   o Round Trip Time (RTT)

   This is the response time between the submission of the ICMP echo
   request packet and the reception of the ICMP echo reply packet. The
   metric can also be regarded as the round-trip delay time. This is
   measured by the ping command. We take the min/avg/max time and the
   loss rate based on measuring this metrics one hundred times by the
   measurement task. An example of the command MAs execute is as
   follows.

   MA-ID $ ping -i 0.05 -c 100 {contents_ip_address}

   o Hop Count and Network Path

This metric refers to the number of intermediate devices (like
routers) through which the data must pass between the MA and the
Target Contents. This metric is regarded as the network distance
between the MA and the Target Contents. This is measured by the
traceroute command. Actually, MA submits the ICMP echo requests
three times. Afterwards, by checking the hop counts and network
path, we can find the change of the network routing on the Internet.
An example of the command MAs execute is as follows.

MA-ID $ traceroute -nIq 3 {contents_ip_address}

o Throughput

This metric refers to how much data can be transferred from the MA to
the Target Contents in a given amount of time. This is regarded as
the bandwidth. We can understand how fast we can get the contents
the on the network. Currently, this is measured by the wget command.
A MA receives URL of the measurement targets and start downloading
the contents using HTTP GET. When the download is completed, the
value divided the contents size by the download complete time is
regarded as the performance metrics of throughput. An example of the
command MAs execute is as follows.

MA-ID $ wget -T 300 -dvO /dev/null {wgetopts} {contents_url}

In addition to above three performance metrics, we are studying the
change of destination IP address of the Internet contents distributed
by some contents service providers. It is important for ISPs to know
the mechanism of the contents delivery networks. MAs resolve some
FQDN and gets the destination IP address. We are studying the
mechanism of the contents delivery networks based on the response
results.

4.2. Measurement Target Contents

The selection of the Target Contents is important for the Internet
measurement; the type, the length, and the number of the contents.
We need to measure the representative contents on the Internet. In
order to find such contents, we have selected contents based on two
viewpoints.

One is the volume of transferred data of network traffic in an ISP.
We obtained partial traffic data on multiple prefectures in Japan.
We selected the Target Contents which were higher in the transferred
traffic volume ranking. Examples of such target Contents are Youtube
Video Streaming files and Mac OS update file on AKAMAI and so on.

Second is the number of access to the contents in the Internet. For
example, the portal sites such as Google or Yahoo!, etc. and the
shopping sites such as Amazon and iTunes, etc. are always the higher
in the number of access to the contents in the Internet.

In another viewpoint, we need to change the target contents according
to the purpose of analysis. If our purpose is to measure an event
traffic, e.g., the download traffic concerning iOS update or the
access traffic concerning the special winning sale of the
professional baseball team, etc., we need to measure the related
contents.

4.3. Measurement Schedule

On receiving the measurement tasks from Controller Server, MAs start
measurement tasks. MAs used to perform the measurement task by
thirty minutes. When the measurement completes, MAs wait the next
scheduled time and do not perform the next measurement tasks. On the
contrary, when the measurement does not complete before the next
scheduled time, the MA kills the measurement process and moves to the
next measurement. The current system in that point is flexible
because the MA can start the next task as soon as a measurement task
is completed. We can collect more kinds of data than before.

4.4. Applications of Measurement

Using the data collected by our measurement system we have studied
how to comprehend and analyze end-to-end performance more accurately
than ever before. A example of analysis is the difference of network
performance between Japanese ISPs, based on the combinations of
Target Contents, measurement time, and areas. We took the
measurement results in consideration to ISP network design and ISP
operation as a reference information.

Furthermore, we have studied the analysis method based on the
combination of our customer feedbacks, remarks on social network
service, and customer voices on our callcenters in addition to the
collected measurement data. As the result of combined analysis, we
expect to find new and useful knowledge we have never found before.

5. Issues of Internet Measurement System

We introduce the issues of Internet Measurement System we have been
operating in this section. The issue section is divided into three
parts: Architecture Issue, Operation Issue, and Security Issue.

5.1. Architecture Issue

   o Scalability

   The Controller Server is connected to receive HTTP GET requests from
   multiple MAs. This means that the Controller Server needs to process
   as many HTTP GET requests as the number of MAs. The number of MAs
   can easily grow beyond the number of HTTP GET requests that a
   Controller Server can process. If we place hundreds of MAs all over
   Japan, we will need to improve the scalability of our system.

   o IPv6 Support

   IPv6 network is constructed totally independently from IPv4 network.
   Hence, the performance of the IPv6 network is highly likely different
   from that of the IPv4 network.

   Although the IPv6 network is not the majority yet, it is growing.
   NTT EAST and WEST provided only 2.7% in NGN (Next Generation Network)
   on December 2013. The rate of IPv6 enabled network in Japan is 27%
   in June 2014[IPv6-Promotion Council]. NTT EAST and WEST presented
   the IPv6 support in PPPoE connection on March 2014. All CPE devices
   for NTT access line already support IPv6 tunneling, allowing users to
   adapt IPv6 easily.

   In order to achieve the broad applicability of our measurement
   results, we will need to investigate the IPv6 performance also.

   o Data Reliability

   We need many kinds of data in order to improve the reliability of
   data analysis. If there are many kinds of data, the reliability of
   our analysis results will be improved and the analysis results might
   be statistically significant. We need to develop the architecture to
   collect as many different types data as possible. When a MA
   completed all the instructed measurement tasks, we are creating the
   measurement system that the MA performs other measurement tasks being
   high priority as soon as possible when a MA completed all the
   instructed measurement tasks.

5.2. Operation Issue

   o Selection of the Products as Measurement Agent

   We used a Japanese product, called OpenBlocks. However, some issues
   happen by using this product. The product sometimes generates high
   heat (e.g., a certain hot day). The heat is hot enough to feel like
   getting burned. In fact, the maximum degree of the heat reaches

about eighty degrees Celsius. Some people were afraid to set up MA
in their house. We are looking for more efficient product satisfying
our requirements than this model.

o   Selection of Measurement Target Contents

It is difficult to decide what contents should be measured to present
the representative performance. There are many kinds of contents on
the Internet.

This time we have selected the Target Contents based on the volume of
transferred data at some points in an ISP. However, there are more
metrics to consider, such as the number of accesses to that contents,
rather than the transferred volume. Other metrics are not studied in
this document.

o   Stable Operation

We had experiences where the measurement results were not sent
immediately, and the measurements for some Target Contents were
failed. Although the actual causes of these difficulties vary (e.g.,
accidentally disconnected LAN cable or power cable), we could easily
respond to those issues using informations (e.g., time and place)
contained in the centralized logs in the Collector Server. Another
difficulty is the change in the settings of the contents provider.
For example, wget command for a video content has not worked due to a
change in a setting in the contents provider. This issue is
difficult to tackle and is left for future work.

5.3. Security Issue

o   Measurement between ISPs and CSPs

If we continuously measure the performance about the contents in the
Internet, from the point of the contents service provider, it can be
obstacle to provide the stable service due to the traffic volume for
measurement. However, we need real situations for the customer to
measure the performance as correctly as possible. The service
provider may request the limitation, e.g., volume and the number of
access, for measurement to MA. In terms of the combination between
ISPs and CSPs, we may need the condition for measurement.

o   DDoS Attack

We placed approximately 150 MAs all over Japan. These MAs may become
DDoS attackers by wrong commands from the Controller Server. From
this reason, the list of commands MAs can perform should be
restricted. And also, the MAs must deny illegal accesses and logins.

MAs should permit only access through instruction from the Controller
Server.

6. Security Considerations

As described in Setion 5.3, security consideration for Internet
measurement must be considered.

7. IANA Considerations

No need to describe any request regarding number assignment.

8. References

8.1. Normative References

[RFC7536] Linsner, M., Eardley, P., Burbridge, T.,and Sorensen, F.,
           "Large-Scale Broadband Measurement Use Cases, May 2015", .

[I-D.deng-lmap-collaboration]
           Deng., L., Huang, R.,and Duan, S., "Use-cases for
           Collaborative LMAP, draft-deng-lmap-collaboration-05 (work
           in progress), June 2015", .

[I-D.ietf-lmap-framewark]
           Eardley, P., Morton, A., Bagnulo, M., Burbridge, T.,
           Aitken, P., and Akhter, P., "A framework for large-scale
           measurement platforms(LMAP),draft-ietf-lmap-framework-14
           (work in progress), April 2015", .

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

8.2. URL References

[MIC]   MIC Report, "http://www.soumu.go.jp/main_sosiki/kenkyu/
           speed_measurement/", .

[sandvine]
           Sandvine Report, "https://www.sandvine.com/pr/2014/5/14/
           sandvine- report-netflix%E2%80%99s-british-invasion.html",
           .

[google]   Google Report,
           "http://www.google.com/get/videoqualityreport/", .

   [plathome]
             OpenBlocks,
             "http://openblocks.plathome.com/products/ax3/", .

   [IPv6-Promotion-Council]
             Japanese IPv6-Promotion,
             "http://v6pc.jp/jp/spread/ipv6spread_02.phtml", .

Authors' Addresses

   Motoyuki Ooki
   NTT Communications
   GranPark 16F
   3-4-1 Shibaura, Minato-ku, Tokyo
   108-8118,Japan

   EMail: m.ooki@ntt.com

   Satoshi Kamei
   NTT Communications
   GranPark 16F
   3-4-1 Shibaura, Minato-ku, Tokyo
   108-8118,Japan

   EMail: skame@nttv6.jp

                A YANG based Data Model for the LMAP Controller
                    draft-oslebo-lmap-control-yang-01.txt

Abstract

   This document defines a YANG data model for the LMAP controller. The
   measurement agents can poll configuration data from the controller
   using RESTCONF.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   The LMAP framework[I-D.ietf-lmap-framework] defines three main
   components:

   o Measurement Agents (MAs) - initiates the actual measurements.

   o Controller - controls one or more MAs. It instructs them when to
      do measurements and when and how to report the results.

   o Collector - accepts measurement results from MAs

   This document defines a YANG data model for the LMAP controller. The
   model can be used by operators to configure all the MAs that are
   controlled by the controller and by MAs to retrieve the configuration
   data. The MAs will poll the information stored on the controller
   using the RESTCONF protocol[I-D.ietf-netconf-restconf]. A future
   draft will define a YANG model for the collector that can be used by
   the MAs to report the results.

   RESTCONF is a protocol currently being standardized in the NETCONF
   working group. It is a HTTP based protocol that defines a standard
   mechansism for accessing data defined in YANG, using datastores
   defined in NETCONF. All the basic features needed for the LMAP
   control protocol is covered by RESTCONF:

   o Full support for Create, Retrieve, Update and Delete (CRUD)
      operations.

   o Support for both XML and JSON format.

o Formal validation based on YANG model.

o Filter mechanisms to request a subset of the target resource
   contents.

o Notification support for controller initiated messages

1.1. Requirements notation

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 [RFC2119].

1.2. Tree Diagrams

A simplified graphical representation of the data model is used in
this document. The meaning of the symbols in these diagrams is as
follows:

o Brackets "[" and "]" enclose list keys.

o Abbreviations before data node names: "rw" means configuration
   (read-write), and "ro" means state data (read-only).

o Symbols after data node names: "?" means an optional node, "!"
   means a presence container, and "*" denotes a list and leaf-list.

o Parentheses enclose choice and case nodes, and case nodes are also
   marked with a colon (":").

o Ellipsis ("...") stands for contents of subtrees that are not
   shown.

2. Data Model Overview

The data model first defines a set of common objects for tasks,
channels, schedules, and timing definitions that are shared among all
the MAs that are controlled by the controller. It then defines a
list of MA objects that contains configuration specific for each MA.
In the MA object there is a list of references to the common
schedules and each reference represents an active schedule for this
specific MA.

The model also contains a notification that tells the MA to contact
the controller and download new configuration.

```
module: lmap-control
   +--rw lmap-control
```

```
      +--rw tasks
      |  +--rw task* [name]
      |     +--rw name                  string
      |     +--rw registry-entry        inet:uri
      |     +--rw role?                 string
      |     +--rw options
      |     |  +--rw option* [name]
      |     |     +--rw name    string
      |     |     +--rw value*  string
      |     +--rw suppress-by-default?   boolean
      |     +--rw cycle-id?             string
      +--rw channels
      |  +--rw channel* [name]
      |     +--rw name          string
      |     +--rw target        inet:uri
      |     +--rw credentials   string
      |     +--rw interface?    leafref
      +--rw schedules
      |  +--rw schedule* [name]
      |     +--rw name      string
      |     +--rw tasks
      |     |  +--rw task* [name]
      |     |     +--rw name                  leafref
      |     |     +--rw channels
      |     |     |  +--rw channel*   leafref
      |     |     +--rw downstream-tasks
      |     |        +--rw downstream-task* [schedule]
      |     |           +--rw schedule    leafref
      |     |           +--rw task?       leafref
      |     |           +--rw outputs
      |     |              +--rw output*   int32
      |     +--rw timing?   leafref
      +--rw timings
      |  +--rw timing* [name]
      |     +--rw name                string
      |     +--rw (timing-type)
      |     |  +--:(periodic)
      |     |  |  +--rw periodic
      |     |  |     +--rw interval    uint32
      |     |  |     +--rw start?      yang:date-and-time
      |     |  |     +--rw end?        yang:date-and-time
      |     |  +--:(calendar)
      |     |  |  +--rw calendar
      |     |  |     +--rw month*            month
      |     |  |     +--rw weekday*          weekday
      |     |  |     +--rw day-of-months*    int8
      |     |  |     +--rw hour*             int8
      |     |  |     +--rw minute*           int8
```

```
  |     |  |     +--rw second*              int8
  |     |  |     +--rw timezone-offset?    timezone-offset
  |     |  |     +--rw start?              yang:date-and-time
  |     |  |     +--rw end?                yang:date-and-time
  |     |  +--:(one-off)
  |     |  |  +--rw one-off-time     yang:date-and-time
  |     |  +--:(immediate)
  |     |  |  +--rw immediate        empty
  |     |  +--:(startup)
  |     |     +--rw startup          empty
  |     +--rw random-spread?    int32
  +--rw log
  |  +--rw log-entry* [agent-id event-time]
  |     +--rw agent-id      leafref
  |     +--rw event-time    int64
  |     +--rw code          int32
  |     +--rw description    string
  +--rw measurement-agents
     +--rw ma* [agent-id]
        +--rw agent-id                      yang:uuid
        +--rw config
        |  +--rw device-id?                        inet:uri
        |  +--rw credentials                       string
        |  +--rw group-id?                         string
        |  +--rw report-id-flag?                   boolean
        |  +--rw control-channel-failure-threshold?    int32
        +--rw schedules
        |  +--rw schedule*    leafref
        +--rw suppressions
        |  +--rw enabled?            boolean
        |  +--rw stop-ongoing-tasks?    boolean
        |  +--rw start?              yang:date-and-time
        |  +--rw stop?               yang:date-and-time
        |  +--rw tasks
        |  |  +--rw task*    leafref
        |  +--rw schedules
        |     +--rw schedule*    leafref
        +--rw status-and-capabilities
           +--rw hardware?          string
           +--rw firmware?          string
           +--rw version?           string
           +--rw interfaces
           |  +--rw interface*    leafref
           +--rw last-measurement?    yang:date-and-time
           +--rw last-report?         yang:date-and-time
           +--rw last-instruction?    yang:date-and-time
           +--rw last-configuration?    yang:date-and-time
           +--rw supported-tasks
```

```
                   |  +--rw supported-task* [name]
                   |     +--rw name        string
                   |     +--rw registry    inet:uri
                   |     +--rw role?       string
                   +--rw conditions
                      +--rw condition* [code]
                         +--rw code    string
                         +--rw text?   string
   notifications:
      +---n update-config
```

3. YANG module

   <CODE BEGINS> file "lmap-control@2014-10-27.yang

   module lmap-control {

     namespace "urn:ietf:params:xml:ns:yang:lmap-control";

     prefix "lmap-control";

     import ietf-yang-types {
       prefix "yang";
     }

     import ietf-inet-types {
       prefix "inet";
     }

     import ietf-interfaces {
       prefix "if";
     }

     organization
       "IETF Large-Scale Measurement Platforms Working Group";

     contact
       "WG Web: <http://tools.ietf.org/wg/lmap/> WG List:
        <mailto:lmap@ietf.org> Editor: Arne Oslebo
        <arne.oslebo@uninett.no>";

     description
       "This YANG module defines a data model for the control protocol
        for Large-Scale Measurement Platforms (LMAP). The information is
        stored on a LMAP controller and Measurement Agents will pull
        this information from the controller at predefined intervals.";

```
     revision 2014-09-09 {
       description
         "Initial revision.";
       reference
         "RFC XXX: A YANG Data Model for the LMAP control protocol";
     }

     typedef weekday {
       type enumeration {
         enum sunday {
           description
             "Sunday of the week";
         }
         enum monday {
           description
             "Monday of the week";
         }
         enum tuesday {
           description
             "Tuesday of the week";
         }
         enum wednesday {
           description
             "Wednesday of the week";
         }
         enum thursday {
           description
             "Thursday of the week";
         }
         enum friday {
           description
             "Friday of the week";
         }
         enum saturday {
           description
             "Saturday of the week";
         }
       }
       description
         "A type modeling the weekdays in the Greco-Roman tradition.";
     }

     typedef month {
       type enumeration {
         enum january {
           description
             "January of the Julian and Gregorian calendar";
         }
```

```
        enum february {
          description
            "February of the Julian and Gregorian calendar";
        }
        enum march {
          description
            "March of the Julian and Gregorian calendar";
        }
        enum april {
          description
            "April of the Julian and Gregorian calendar";
        }
        enum may {
          description
            "May of the Julian and Gregorian calendar";
        }
        enum june {
          description
            "June of the Julian and Gregorian calendar";
        }
        enum july {
          description
            "July of the Julian and Gregorian calendar";
        }
        enum august {
          description
            "August of the Julian and Gregorian calendar";
        }
        enum september {
          description
            "September of the Julian and Gregorian calendar";
        }
        enum october {
          description
            "October of the Julian and Gregorian calendar";
        }
        enum november {
          description
            "November of the Julian and Gregorian calendar";
        }
        enum december {
          description
            "December of the Julian and Gregorian calendar";
        }
      }
      description
        "A type modeling the month in the Julian and Gregorian
         tradition.";
```

```
      }

      typedef timezone-offset {
        type string {
          pattern "Z|[\+\-]\d{2}:\d{2}";
        }
        description
          "A timezone-offset as it is use in the yang:date-and-time type.
           The value Z is equivalent to +00:00. The value -00:00
           indicates and unknown time-offset.";
      }

      grouping timing-start-end-grouping {
        description
          "A grouping that provides start and end times for timing
           objects.";
        leaf start {
          type yang:date-and-time;
          description
            "The date and time when the timing object starts to create
             triggers.";
        }
        leaf end {
          type yang:date-and-time;
          description
            "The date and time when the timing object stops to create
             triggers. It is generally a good idea to always configure an
             end time and to refresh the configuration of timing object
             as needed to ensure that agents that loose connectivity to
             their controller do not continue their tasks forever.";
        }
      }

      container lmap-control {
        description
          "LMAP Control configuration";
        container tasks {
          description
            "Task configuration";
          list task {
            key "name";
            description
              "List of available tasks.";
            leaf name {
              type string;
              mandatory "true";
              description
                "The unique name of the task.";
```

```
              }
              leaf registry-entry {
                type inet:uri;
                mandatory "true";
                description
                  "Registry URI identifying the task.";
              }
              leaf role {
                type string;
                description
                  "Measurement Agent role.";
              }
              container options {
                description
                  "Configuration of options passed to the task.";
                list option {
                  key "name";
                  description
                    "List of options.";
                  leaf name {
                    type string;
                    description
                      "Option name.";
                  }
                  leaf-list value {
                    type string;
                    description
                      "Option value.";
                  }
                }
              }
              leaf suppress-by-default {
                type boolean;
                default "true";
                description
                  "Indicates the behavior of the default suppress
                   instruction for this task. If FALSE then the Task will
                   not be suppressed.";
              }
              leaf cycle-id {
                type string;
                description
                  "Measurement Cycle ID that can be used to easily identify
                   a set of measurement results. The ID should be
                   incremented or changed when a task option is changed so
                   that two sets of results can no longer be directly
                   compared.";
              }
```

```
          }
        }
        container channels {
          description
            "Channel configuration";
          list channel {
            key "name";
            description
              "The list of available channels";
            leaf name {
              type string;
              description
                "The unique name of the channel";
            }
            leaf target {
              type inet:uri;
              mandatory "true";
              description
                "The remote endpoint of the channel.";
            }
            leaf credentials {
              type string;
              mandatory "true";
              description
                "The security credentials to establish a secure
                 channel.";
            }
            leaf interface {
              type leafref {
                path "/if:interfaces/if:interface/if:name";
              }
              description
                "Name of interface to use when MA has multiple
                 interfaces.";
            }
          }
        }
        container schedules {
          description
            "Schedules configuration";
          list schedule {
            key "name";
            description
              "The list of available schedules.";
            leaf name {
              type string;
              description
                "The unique name of the schedule.";
```

```
            }
          container tasks {
            description
              "The configuration of tasks that are executed as part of
               this schedule.";
            list task {
              key "name";
              description
                "The list of the tasks.";
              leaf name {
                type leafref {
                  path "/lmap-control/tasks/task/name";
                }
                description
                  "The reference to a named task that will be
                   executed";
              }
              container channels {
                description
                  "Configuration of channels used by the task.";
                leaf-list channel {
                  type leafref {
                    path "/lmap-control/channels/channel/name";
                  }
                  description
                    "List of channels used by the task.";
                }
              }
              container downstream-tasks {
                description
                  "Configuration of downstream tasks. The output from
                   this task is sendt to the selected task.";
                list downstream-task {
                  key "schedule";
                  description
                    "";
                  leaf schedule {
                    type leafref {
                      path "/lmap-control/schedules/schedule/name";
                    }
                    description
                      "Name of downstream schedule.";
                  }
                  leaf task {
                    type leafref {
                      path "/lmap-control/schedules/"
                        + "schedule[name=current()/../schedule]/"
                        + "tasks/task/name";
```

```
                      }
                      description
                        "Name of downstream task that is part of the
                         above downstream schedule.";
                    }
                  container outputs {
                    description
                      "Configuration of task outputs";
                    leaf-list output {
                      type int32;
                      description
                        "List of task outputs that will be sendt to
                         this downstream task.";
                    }
                  }
                }
              }
            }
          }
          leaf timing {
            type leafref {
              path "/lmap-control/timings/timing/name";
            }
            description
              "The timing of the schedule.";
          }
        }
      }
      container timings {
        description
          "Configuration of LMAP timings. Implementations may be forced
           to delay acting upon triggers in the face of local
           constraints. A task triggered therefore not rely on the
           accuracy provided by the scheduler implementation.";
        list timing {
          key "name";
          description
            "The list of timings configured on the LMAP agent.";
          leaf name {
            type string;
            description
              "The unique name of a timing.";
          }
          choice timing-type {
            mandatory "true";
            description
              "Different types of timing objects are handled by
               different branches of this choices.";
```

```
            case periodic {
              container periodic {
                description
                  "A periodic timing object triggers periodically
                   driven by a regular interval.";
                leaf interval {
                  type uint32;
                  units "milliseconds";
                  mandatory "true";
                  description
                    "The number of milliseconds between two triggers
                     generated by this periodic timing object. The
                     execution system must not generate triggers for
                     periodic timing objects that have a interval value
                     of 0. A timing object with an interval of 0
                     milliseconds will therefore never trigger.";
                }
                uses timing-start-end-grouping;
              }
            }
            case calendar {
              container calendar {
                description
                  "A calendar timing object trigger based on the
                   current calendar date and time.";
                leaf-list month {
                  type month;
                  description
                    "A month at which this calendar timing will
                     trigger.";
                }
                leaf-list weekday {
                  type weekday;
                  description
                    "A weekday at which this calendar timing will
                     trigger.";
                }
                leaf-list day-of-months {
                  type int8 {
                    range "-31..-1 | 1..31";
                  }
                  description
                    "A day in the months at which this calendar timing
                     will trigger. Negative numbers indicate days
                     counted backwards from the end of the months.";
                }
                leaf-list hour {
                  type int8 {
```

```
              range "0..23";
            }
            description
              "An hour at which this calendar timing will
               trigger.";
          }
          leaf-list minute {
            type int8 {
              range "0..59";
            }
            description
              "A minute at which this calendar timing will
               trigger.";
          }
          leaf-list second {
            type int8 {
              range "0..59";
            }
            description
              "A second at which this calendar timing will
               trigger.";
          }
          leaf timezone-offset {
            type timezone-offset;
            description
              "The timezone in which this calendar timing object
               will be evaluated.";
          }
          uses timing-start-end-grouping;
        }
      }
      case one-off {
        leaf one-off-time {
          type yang:date-and-time;
          mandatory "true";
          description
            "This one-off timing object triggers once at the
             configured one-off-time.";
        }
      }
      case immediate {
        leaf immediate {
          type empty;
          mandatory "true";
          description
            "This immediate timing object triggers immediately
             when it is configured.";
        }
```

```
              }
              case startup {
                leaf startup {
                  type empty;
                  mandatory "true";
                  description
                    "This startup timing object triggers whenever the
                     LMAP agent (re)starts.";
                }
              }
            }
            leaf random-spread {
              type int32;
              units "milliseconds";
              description
                "This optional leaf adds a random spread to the
                 computation of the trigger.";
            }
          }
        }
        container log {
          description
            "";
          list log-entry {
            key "agent-id event-time";
            description
              "";
            leaf agent-id {
              type leafref {
                path "/lmap-control/measurement-agents/ma/agent-id";
              }
              description
                "";
            }
            leaf event-time {
              type int64;
              description
                "Timestamp (number of ms since January 1, 1970, 00:00:00
                 GMT) of the log event";
            }
            leaf code {
              type int32;
              mandatory "true";
              description
                "";
            }
            leaf description {
              type string;
```

```
              mandatory "true";
              description
                "";
            }
          }
        }
        container measurement-agents {
          description
            "Configuration of Measurement Agents.";
          list ma {
            key "agent-id";
            description
              "List of Mesurement Agents that are controlled by a
               controller.";
            leaf agent-id {
              type yang:uuid;
              mandatory "true";
              description
                "The unique ID of the MA";
            }
            container config {
              description
                "Configuration of a Measurement Agent.";
              leaf device-id {
                type inet:uri;
                description
                  "Optional device ID that can be used to identify the MA
                   before an agent ID is assigned.";
              }
              leaf credentials {
                type string;
                mandatory "true";
                description
                  "The credentials of the agent.";
              }
              leaf group-id {
                type string;
                description
                  "Optional group ID of the agent";
              }
              leaf report-id-flag {
                type boolean;
                default "false";
                description
                  "If true then the agent ID will be reported to the
                   collectors.";
              }
              leaf control-channel-failure-threshold {
```

```
                  type int32;
                  description
                    "";
                }
              }
            container schedules {
              description
                "Configuration of active schedules for this MA.";
              leaf-list schedule {
                type leafref {
                  path "/lmap-control/schedules/schedule/name";
                }
                description
                  "List of active schedules.";
              }
            }
            container suppressions {
              description
                "Configuration of suppressions";
              leaf enabled {
                type boolean;
                default "false";
                description
                  "Setting this to true will suppress all tasks where
                   suppress-by-default is true.";
              }
              leaf stop-ongoing-tasks {
                type boolean;
                default "false";
                description
                  "Setting this to true will terminate all running tasks
                   when enabled is also set to true.";
              }
              leaf start {
                type yang:date-and-time;
                description
                  "The date and time when suppression should start. If
                   not present then suppression will start
                   immediately.";
              }
              leaf stop {
                type yang:date-and-time;
                description
                  "The date and time when suppression should stop. If not
                   present the suppression will continue until 'enabled'
                   is set to false.";
              }
              container tasks {
```

```
              description
                "Configuration of tasks that will be suppressed.";
              leaf-list task {
                type leafref {
                  path "/lmap-control/tasks/task/name";
                }
                description
                  "The list of tasks that will be suppressed. If the
                   list is empty, then all tasks will be suppressed.";
              }
            }
            container schedules {
              description
                "Configuration of schedules that will be suppressed.";
              leaf-list schedule {
                type leafref {
                  path "/lmap-control/schedules/schedule/name";
                }
                description
                  "The list of schedules that will be suppressed. If
                   the list is empyt, then all schedules will be
                   suppressed.";
              }
            }
          }
          container status-and-capabilities {
            description
              "Operational status and capabilities of the measurement
               agent.";
            leaf hardware {
              type string;
              default "true";
              description
                "A short description of the hardware that the
                 measurement agent is running on.";
            }
            leaf firmware {
              type string;
              description
                "The firmware version that the measurement agent is
                 running on";
            }
            leaf version {
              type string;
              description
                "The version number of the measurement agent
                 software.";
            }
```

```
            container interfaces {
              description
                "Interfaces available for measurements.";
              leaf-list interface {
                type leafref {
                  path "/if:interfaces/if:interface/if:name";
                }
                description
                  "List of interfaces";
              }
            }
            leaf last-measurement {
              type yang:date-and-time;
              description
                "The date and time of the last performed measurement
                 task.";
            }
            leaf last-report {
              type yang:date-and-time;
              description
                "The date and time of the last reporting task.";
            }
            leaf last-instruction {
              type yang:date-and-time;
              description
                "The date and time for when the measurement agent last
                 received new instructions.";
            }
            leaf last-configuration {
              type yang:date-and-time;
              description
                "The date and time for when the measurement agent last
                 received new configuration data.";
            }
            container supported-tasks {
              description
                "Supported tasks.";
              list supported-task {
                key "name";
                description
                  "The list of tasks supported by the measurement
                   agent.";
                leaf name {
                  type string;
                  description
                    "The name of the supported task.";
                }
                leaf registry {
```

```
                      type inet:uri;
                      mandatory "true";
                      description
                        "The registry entry of the supported task.";
                  }
                  leaf role {
                    type string;
                    description
                      "The role of the measurement agent in the in the
                       supported task.";
                  }
                }
              }
              container conditions {
                description
                  "";
                list condition {
                  key "code";
                  description
                    "";
                  leaf code {
                    type string;
                    description
                      "";
                  }
                  leaf text {
                    type string;
                    description
                      "";
                  }
                }
              }
            }
          }
        }
      }

    notification update-config {
      description
        "Notification telling the MA to contact the controller and
         download new configuration.";
    }
  }


  <CODE ENDS>
```

4. Security Considerations

    TBD

5. Acknowledgements

    Some parts of the YANG model was copied from [I-D.schoenw-lmap-yang].

6. 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-08 (work in progress), August 2014.

    [I-D.ietf-netconf-restconf]
              Bierman, A., Bjorklund, M., Watsen, K., and R. Fernando,
              "RESTCONF Protocol", draft-ietf-netconf-restconf-01 (work
              in progress), July 2014.

    [I-D.schoenw-lmap-yang]
              Schoenwaelder, J. and V. Bajpai, "A YANG Data Model for
              LMAP Measurement Agents", draft-schoenw-lmap-yang-01 (work
              in progress), September 2014.

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

Appendix A. Open issues

    The traditional usage of both NETCONF and RESTCONF is to push
    configuration to the managed agents. Using the proposal described in
    this draft the operator will push the configuration to the LMAP
    controller, but the MAs will pull the configuration from the
    controller. At the interim meeting in Dublin September 15 some
    concern was raised for using RESTCONF in a scenario like this. The
    question was posted to the NETCONF mailing list and one answer said
    that the preferred way would be to use the call-home strategy of TLS.
    The reason for this is to preserve the direction of the
    authentication mechanisms. So far it has been decided not to include
    any call-home functionality in RESTCONF, only in NETCONF.

Appendix B. Example configuration

       <lmap-control>
         <tasks>
           <task>

```
      <name>Configuration</name>
      <registry-entry>urn:..:configuration</registry-entry>
        </task>
        <task>
      <name>Report</name>
      <registry-entry>urn:..:report</registry-entry>
        </task>
        <task>
      <name>Ping</name>
      <options>
        <option>
          <name>destination</name>
          <value>127.0.0.1</value>
        </option>
      </options>
      <registry-entry>urn:ping</registry-entry>
        </task>
      </tasks>
      <channels>
        <channel>
      <name>Control channel</name>
      <target>http://www.example.com:5000</target>
      <credentials></credentials>
        </channel>
        <channel>
      <name>Report channel</name>
      <target>http://www.example.com:5001</target>
      <credentials></credentials>
        </channel>
      </channels>
      <schedules>
        <schedule>
      <name>Configuration</name>
      <tasks>
        <task>
          <name>Configuration</name>
          <channels>
            <channel>Control channel</channel>
          </channels>
        </task>
      </tasks>
      <timing>Hourly</timing>
        </schedule>
        <schedule>
      <name>Ping</name>
      <tasks>
        <task>
          <name>Ping</name>
```

```
      <downstream-tasks>
        <downstream-task>
      <schedule>Report daily</schedule>
      <task>Report</task>
      <outputs>
        <output>1</output>
      </outputs>
        </downstream-task>
      </downstream-tasks>
    </task>
  </tasks>
  <timing>Hourly</timing>
    </schedule>
    <schedule>
  <name>Report daily</name>
  <tasks>
    <task>
      <name>Report</name>
      <channels>
        <channel>Report channel</channel>
      </channels>
    </task>
  </tasks>
  <timing>Daily</timing>
    </schedule>
  </schedules>

  <timings>
    <timing>
  <name>Hourly</name>
  <periodic>
    <interval>60000</interval>
  </periodic>
    </timing>
    <timing>
  <name>Daily</name>
  <periodic>
    <interval>1440000</interval>
  </periodic>
    </timing>
  </timings>

  <measurement-agents>
    <ma>
  <agent-id>00000000-0000-0000-0000-000000000001</agent-id>
  <config>
    <device-id>Device ID 1</device-id>
    <credentials></credentials>
```

```
      <group-id>1</group-id>
      <control-channel-failure-threshold>
        5
          </control-channel-failure-threshold>
    </config>
    <schedules>
      <schedule>Configuration</schedule>
    </schedules>
      </ma>
      <ma>
    <agent-id>00000000-0000-0000-0000-000000000002</agent-id>
    <config>
      <device-id>Device ID 2</device-id>
      <credentials></credentials>
      <group-id>1</group-id>
      <control-channel-failure-threshold>
        5
      </control-channel-failure-threshold>
    </config>
    <schedules>
      <schedule>Configuration</schedule>
      <schedule>Ping</schedule>
    </schedules>
      </ma>
    </measurement-agents>
  </lmap-control>
```

Appendix C. RESTCONF queries

C.1. Get configuration - XML format

   This example shows how a MA can retrieve the configuration
   information from the controller in XML format. Note that some lines
   are wrapped for display purposes only.

   GET /restconf/data/lmap-control:lmap-control/measurement-agents/
   ma=00000000-0000-0000-0000-000000000001/config HTTP/1.1
   Host: 127.0.0.1:5000
   Accept: application/yang.data+xml,
       application/yang.errors+xml

   The server might respond as follows.

```
HTTP/1.0 200 OK
Content-Type: application/yang.data+xml
Server: example-server
Date: Fri, 12 Sep 2014 11:56:42 GMT

<config>
  <credentials/>
  <group-id>1</group-id>
</config>
```

C.2. Get schedules - JSON format

This example shows how a MA can retrieve a list of active schedules from the controller in JSON format.

```
GET /restconf/data/lmap-control:lmap-control/measurement-agents/
ma=00000000-0000-0000-0000-000000000002/schedules HTTP/1.1
Host: 127.0.0.1:5000
Accept: application/yang.data+json,
    application/yang.errors+json
```

The server might respond as follows.

```
HTTP/1.0 200 OK
Content-Type: application/yang.data+json
Server: example-server
Date: Fri, 12 Sep 2014 12:14:15 GMT

{
    "schedules": {
        "schedule": [
            "Configuration",
            "Ping"
        ]
    }
}
```

Author's Address

   Arne Oslebo
   UNINETT

   Email: arne.oslebo@uninett.no