Network Working Group
Internet Draft
Intended status: Standards Track
Expires: November 7, 2015

V.Liu
L.Deng
ChinaMobile

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

Copyright Notice

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