From global measurements to local management

D 1.1: Initial architecture for Leone's network management framework, including goals and constraints

Date: 31st August 2013

Version: 1

Dissemination level: Public (PU)

Project co-funded by Leone Partners and by the European Commission in the 7th Framework Programme

ABSTRACT

This document presents an initial proposal for an architecture for large-scale measurements to enhance network management. The architecture has been developed by the Leone collaborative research project.

The key challenges identified include scaling, as we envisage a measurement agent in every home gateway and edge device, and flexibility that allows new tests to be added and the management system to readily adjust the schedule of tests.

The architecture is currently in its initial form, so this document also highlights some of the open research issues. Validation activity includes our internal implementation and testing of various components, our trial and our discussions with external people, for example in standards bodies. Indeed, standardisation is a key activity, so that the measurement capability is more pervasive and manageable and the performance metrics are directly comparable.

The intended readership for this report is quite general – other network management researchers, protocol designers, network management architects and engineers. Feedback is very welcome – indeed, it forms an important part of validating that the architecture is reasonable.

ABOUT THE PROJECT

Project Coordinator: Philip Eardley, BT

Technical Manager: Sam Crawford, SamKnows

Start date of project: November 1st 2012

Duration: 30 months

Acknowledgement: Leone is a Collaborative project under FP7 ICT-2011.1.1 Future Networks. The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement n° 317647.

ABOUT THE DOCUMENT

Editor: Philip Eardley (BT)

Due date of deliverable: 31st August 2013

Actual submission date: 31st August 2013

Disclaimer

This document contains material, which is the copyright of certain Leone consortium parties, and may not be reproduced or copied without permission.

The commercial use of any information contained in this document may require a license from the proprietor of that information.

Neither the Leone consortium as a whole, nor a certain party of the Leone consortium, warrant that the information contained in this document is capable of use, or that use of the information is free from risk, and accept no liability for loss or damage suffered by any person using this information.

LIST OF AUTHORS

Philip Eardley (BT)
Trevor Burbridge (BT)
Juergen Schoenwaelder (Jacobs)
Vaibhav Bajpai (Jacobs)
Marcelo Bagnulo (UC3M)
Sam Crawford (SamKnows)

EXECUTIVE SUMMARY

Measurements can help a network operator understand the quality experienced by its customers. They allow more effective capacity planning and network design. They help identify problems in the network and with equipment or suppliers, and to isolate whether the issue is in the shared part of the network, unique to a single user line, in the home network or an over-the-top service.

Although there are many measurements of the Internet today, they have been developed, deployed and operate independently, so measurement results are hard to compare and systems are hard to scale and integrate into existing network management. We believe that our architecture for large-scale measurements will enable a more comparable, pervasive and manageable measurement capability so that network management becomes more powerful and cohesive.

By 'architecture' we refer principally to the basic functional components required to make measurements of the network, how to control those measurements and how to collect the results for use by network management systems. The main functions are: Measurement Agents and Measurement Peers which jointly generate test traffic and measurement some metric of interest associated with its transfer (such as 'time to transfer a test file' or 'packet loss'); a Controller which instructs the Measurement Agents about what tests are to be done when and how to report the results; and a Collector which gather the measurement results from the Measurement Agents.

The critical interfaces are between the Controller and Measurement Agent, and the Measurement Agent and Collector. We have made a first proposal for a protocol implementing these interfaces. It is based on HTTP with information encoded in JSON. The approach is motivated by their wide deployment. In order to reach agreement about what information needs to be transferred over the protocol, we are also developing an abstract, protocol-neutral definition of the information; we hope that such an information model could become the single universally-accepted standard, as this would allow some degree of interoperability between our HTTP-based solution and other solutions, for instance perhaps based on the Broadband Forum protocol TR-069.

There are several open issues, for example how to make an 'admission control' check that the test traffic associated with making a measurement won't overload the end user, measurement peer or network, and how to enhance the measurement results with information about the subscriber, such as the Mbit/s rate in their broadband contract.

Standardisation is a key focus of our activity as standards would enable comparability of measurements made of the same metric at different times and places, and would allow the operator of a measurement system to buy the various components from different vendors. We are working with colleagues at the Broadband Forum and IETF to standardise them, in particular by initiating the creation of a new IETF working group called LMAP ('large-scale measurement platforms').

Development of an architecture is essentially an artistic process: it cannot be formally derived from requirements or goals or proven to be the unique solution of an engineering problem. Continuous validation of the design is therefore a critical process. Thus we continue a detailed study of our proposals, through prototyping, our trials and discussion for example in standardisation forums.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

API	Application Programming Interface
CDN	Content Distribution Network
CGNAT	Carrier Grade Network Address Translator
CPE	Customer Premises Equipment
CSV	Comma-Separated Values
DDoS	Distributed Denial of Service
DNS	Domain Name System
DOCSIS	Data Over Cable Service Interface Specification
DSL	Digital Subscriber Line
DTLS	Datagram Transport Layer Security
FQDN	Fully Qualified Domain Name
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IP	Internet Protocol
IPFIX	Internet Protocol Flow Information Export
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
JSON	JavaScript Object Notation
LMAP	Large-scale Measurement Platform
MA	Measurement Agent
NAT	Network Address Translation
NETCONF	Network Configuration Protocol
OAM	Operations, Administration, and Maintenance
OSS	Operations support systems
QoE	Quality of Experience
QoS	Quality of Service
REST	Representational state transfer
SPD	Subscriber Parameter Database

SQL	Structured Query Language
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TR	Technical Report
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UUID	Universally Unique Identifier

1 INTRODUCTION

This document presents the initial results of the Leone project to develop an architecture for network management and in particular for the large-scale measurements that can help network management. We stress that these are interim results produced part way through the first year of a 2.5 year project. The work will be updated in a year or so.

Future network management needs to focus on improving the end user's experience in a world of highly distributed and meshed applications. Compared with today, two innovations seem to be required: a focus on the 'quality of experience' – the performance and functionality experienced by end users – and the ability to integrate multi-dimensional information – measurements, control plane information and so on.

We believe that the right way of addressing the first point is to make measurements from probes at or near the end user, for example embedded in home gateways, since end-to-end measurements are the best way of gaining insight into the user's service. Further, probing to different test points in the network should be able to help network management systems isolate where a problem is in the network and design network upgrades more effectively. Although there are many "measurements" of the Internet today, they have been developed, deployed and operate independently, so measurement results are hard to compare and systems are hard to scale and integrate into existing network management. We believe that an architecture for large-scale measurements enables a more cohesive approach.

The information obtained from such large-scale path measurements can be combined with other types of measurements, and network and operational data. Some of these measurements may be obtained through new standard interfaces (as we discuss later). However, other equipment will continue to report over existing standards such as SNMP or Netflow. Ultimately even a network management system that has access to all the different network measurement and management interfaces will still need to obtain other information such as consumer and product information or data from other network management systems. Such information will come via standard IT integration techniques. These include techniques such as obtaining the data via a database or repository using SQL or NoSQL approaches or through application messages (JMS, MQ) or application interfaces (e.g. Web Services). The IT integration industry is vast and well established and we cannot expect all data to be shared via network management specific protocols once the data is beyond the network elements.

For the above reasons we haven't developed a general architecture for integrating multi-dimensional information. The right approach seems highly dependent on the actual information – it is essentially an application integration problem which anyway is hardly unique to network management. It will also vary from organisation to organisation dependent on their preferred integration strategies (such as enterprise information buses and big data repositories). Instead we are studying a couple of specific examples of integrating measurement information with routing information, as discussed in other Leone documents. The purpose of this is to show what can be done through multi-dimensional data rather than prescribing how the different data sources are integrated.

While we do not believe it is possible to prescribe a single integration approach for multi-dimensional data, we do believe that the large-scale end-to-end measurements themselves should be conducted using standard protocols between components in a standard architecture. By 'architecture' we refer principally to the basic functional components required to make measurements of the network, how to control those measurements and how to collect the results for use by network management systems. The main functions are: Measurement Agents and Measurement Peers which jointly generate test traffic and measurement some metric of interest associated with its transfer (such as 'time to transfer a test file' or 'UDP packet loss'); a Controller which manages the Measurement Agents (MAs); and a Collector to gather the measurement results from the MAs.

We follow a two-stage process to design the Controller-MA and MA-Collector interfaces: an Information Model, which is an abstract, protocol-neutral definition of the information to be transferred to and from the MA; and a Data Model, which encodes the information model into a specific structured format that can then be exchanged using a particular transport protocol.

It is also important that an architecture is clear about the high-level goals – the things it is trying to enable – and the simplifications it is prepared to accept. These impose constraints on the technical solutions and provide a rationale for choices between different options.

The above aspects provide a framework within which other research work of the project can fit, for example about how to measure the parameters of interest, and how the measurement results can be analysed, visualised and integrated with network management.

The intended readership for this report is quite general – other network management researchers, protocol designers, network management architects and engineers. We will continue to seek their feedback through dissemination at conferences, concertation activities especially with the mPlane and Flamingo projects, and standardisation at the IETF and Broadband Forum. The architecture will change as a result of that feedback, as well as our own validation activities which develop and trial technical solutions.

The report summarises our work, whilst the Appendices contains extensive details.

We believe that our architecture will enable a more comparable, pervasive and manageable measurement capability so that network management becomes more powerful and cohesive.

2 GOALS AND USE CASES

An ISP, or indeed another network operator, needs to understand the performance of their networks, the performance of the suppliers (downstream and upstream networks), the performance of services, and the impact that such performance has on the experience of their customers. We have identified several motivations for an ISP to measure performance, in summary:

- Identifying, isolating and fixing problems in the network, services or with CPE and end user equipment. Such problems may be at a shared point in the network topology (e.g. a single exchange), common to a vendor or equipment type (e.g. line card or home gateway) or unique to a single user line (e.g. copper access). For example, when a customer reports a fault, the call centre agent could trigger immediate measurement tests and get the results whilst the customer is still on-line. This would help to identify intermittent faults ("the problem is right now") and would reduce the ISP's burden of re-identifying a fault that impacts many customers.
- Capacity planning and network design. The ISP can monitor network performance indicators and so design and plan their network to ensure specified levels of user experience. As well as deploying more capacity, an operator could move a service closer to end users or assess the impact of QoS. Service Level Agreements may be defined at network or product boundaries.
- Understanding the quality experienced by customers. The ISP can gain better insight into the user's service. The end-to-end perspective matters, across home /enterprise networks, peering points, CDNs etc.
- Understanding the impact and operation of new devices and technology. Performance measurements would enable extensive beta testing of a new technology, product or service before it is deployed for a live service, and a better understanding of its operation and impact on other services. It also helps to quantify the advantage of a new technology and so support the business case for larger roll-out.

Regulators also would like a measurement capability, in order to benchmark the performance of different ISPs. Publicising the results stimulates competition and so pressurises ISPs to improve their broadband service.

End users want to use measurements to run diagnostic checks, for example to see if their network is performing according to their service level agreement. An ISP would also like to distribute a self-help tool that a customer could use to optimise their home wireless network, and to help them identify whether any problem exists with an over-the-top service instead of with their broadband product.

From the above use cases we have identified several goals for our work:

- Standardised: in terms of the tests that they perform, the components, the data models and protocols for transferring information between the components. Today's systems are proprietary in some or all of these aspects. Standards would enable comparability of measurements made of the same metric at different times and places, and would allow the operator of a measurement system to buy the various components from different vendors.
- Extensible: it should be easy to add or modify tests.
- Large-scale: we envisage a Measurement Agent (MA) in every home gateway and edge device such as set-top-boxes and tablet computers. Existing systems have up to a few thousand MAs.
- Diversity: a measurement system should handle different types of Measurement Agent: from different vendors, for wired and wireless, for IPv4 or IPv6, and so on.

3 OVERALL ARCHITECTURE

It is challenging to meet the goals of the use cases outlined in the previous section; a measurement capability is needed that is pervasive, manageable, standardised and provides comparable performance metrics. Rather than dive straight into detailed protocol design, it is better first to consider the overall framework: a definition of the high-level elements and their interactions, and constraints on the detailed design.

The Leone framework has four basic components, as shown in Figure 1: Measurement Agents, Measurement Peers, Controllers and Collectors.

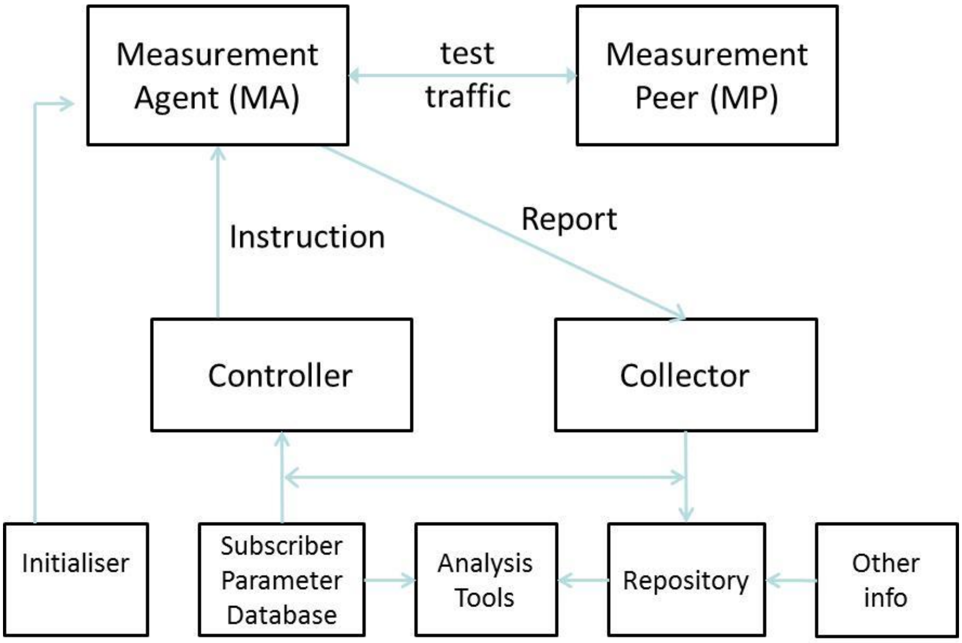

Figure 1: Leone architecture for large-scale measurements

Measurement Agents (MAs) perform active measurements in conjunction with the Measurement Peers (MPs), by generating test traffic and measuring some metric associated with its transfer over the path between the MA to MP; for example the time taken to transfer a 'test file'. A MA can also perform passive measurements, in which case a MP is not required. The architecture does not limit what form the MA can take – it could be a dedicated piece of hardware, a piece of software sitting on a shared device, or anything in between. Section 6 considers a registry for tests.

A Controller manages Measurement Agents. It instructs MAs about what Measurement Tasks to perform, when they should execute, with which parameters and against which MPs. For example it may instruct a MA at a home gateway: "Run the 'download speed test' with the MP at the end user's first IP point in the network; if the end user is active then delay the test and re-try 1 minute later, with up to 3 re-tries; repeat every hour at xx.05". A Controller can also initiate one-off tests, as well as regular ones. It also instructs MAs about when it should report its Measurement Results and where to.

A Collector accepts a Measurement Report from MAs with their Measurement Results. A MA might report its results to several Collectors.

Other components are the Initialiser, which effectively bootstraps the MA so that it can start to communicate with the Controller, and the Subscriber Parameter Database (Section 7), which contains information about the broadband line (such as its contracted rate and time zone) that affects the choice of Measurement Task and the interpretation of the Results.

The Results are combined with other data, analysed and visualised using Data Analysis Tools, integrated with the operator's OAM tools or other third party tools.

Our framework includes various assumptions. These are effectively proposed constraints on the scope of the solution, in order to make the detailed work more tractable.

Our first constraint is that a measurement system must be under the direction of a single organisation. This means that one party is responsible for both data confidentiality/privacy and the impact of tests on users. Clear responsibility is critical, given that a misbehaving large-scale measurement system could potentially harm user experience, user privacy and network security. Our constraint also simplifies the solution as it avoids policy decisions and coordination between measurement systems.

Secondly, each Measurement Agent is only associated with a single Controller at any point in time. The constraint avoids different Controllers giving a MA conflicting instructions and so means that the MA does not have to manage contention between multiple Test (or Report) Schedules. This simplifies the design of MAs, which is critical for a large-scale infrastructure.

The third constraint is that only Measurement Agents - and not Measurement Peers - can initiate Measurement Tasks and communicate with Controllers and Collectors. There are several reasons. MAs have special-purpose measurement functionality, whilst an MP may be a device performing its normal operational role (for example a DNS or web server). Also, an MA will typically be embedded on a home gateway and so behind a NAT; since a MA always initiates a Measurement Task, the MP will naturally learn the MA's public-facing IP address. Further, since the only MAs communicate with the Collectors (not the MP), the reporting process is easier to secure.

There are several open issues about the framework that we are investigating. In summary:

How are the Measurement Results enhanced with information about the subscriber information? The subscriber's particular broadband contract or type of home hub, for example, will impact how the Results are interpreted. We favour the Subscriber Parameter Database directly informing the Data Analysis Tools, rather than sending the information via the MA.

Can the MA negotiate with the Controller about what Measurements Tasks it can run? We believe that negotiation should be avoided (the Controller simply instructs the MA) because negotiation adds considerably complexity to the MA, Controller and Control Protocol, for little benefit.

How do we ensure that the test traffic doesn't interfere with the end user's real traffic, and doesn't overload the Measurement Peer or the network? We believe that at the start of a Measurement Task a MA and MP should be able to do some 'admission control' check if necessary (for example, delaying or rejecting the Measurement Task if the subscriber is active).

How can the Controller instruct the MAs to suspend sending test traffic, for instance if the network is unexpectedly heavily overloaded? This is difficult in the normal situation where the MA sits behind a NAT.

4 INFORMATION MODEL

The goal of the Information Model is to define the information which is held and passed to and from the Measurement Agent. Defining this information serves a number of purposes:

- To form agreement about what information needs to be passed over the MA Control and Reporting interfaces between the MA and the Controller and Collector respectively and to assist in discussion about what needs to be standardised and what is available using existing mechanisms (e.g. clock, security credentials)
- To guide in the standardisation of different protocol and data model implementations of the MA Control and Reporting interfaces
- To enable a very high level interoperability between different Control and Reporting interface protocols. It is possible that a Controller with a single information model and internal data structure could instruct two or more sub-populations of MAs using different protocols. The Controller would perform the appropriate mapping between its own data representation and the data model of each protocol.

The Information Model developed within Leone has been presented in brief to the Broadband Forum and has been submitted as a draft to the IETF LMAP working group. We believe that there should be a single common Information Model so that even if different groups implement different protocols for the MA we can ensure that these diverse MAs can still be controlled by a single Controller and will implement the same control and reporting capabilities – e.g. an IETF and Broadband Forum Measurement Schedule can both specify a calendar-base schedule of the same test with the same configuration parameters.

The Information Model is broken into a number of parts according to their different purposes. Analysis of the LMAP framework reveals that different information needs to be transferred between the various elements, as reflected in the structure shown in Figure 2. For example, the Instruction information is transmitted from the Controller to the MA, the Reporting information is transmitted from the MA to the Collector, whilst the pre-configuration information is configured on the MA before interaction with the Controller. Both the Logging and Status parts are transmitted from the MA to the Controller, but at different times and for different reasons.

Pre-configuration	Minimal set of information necessary for an MA to securely contact an initial Controller
Configuration	Information configured by the Controller pertaining to Controller communication or general MA settings such as MA and Group ID
Instruction	Configuration by the Controller of what Measurement Tasks to perform, when to perform them, and where/when to report the results
Logging	Information transmitted back to the Controller with configuration or instruction errors and general failure notices
Status	Information available to be fetched by the Controller such as the Measurement Tasks supported by the MA
Reporting	Information sent to the Collector regarding the Measurement Task results including MA context and Task Configuration

Figure 2: Overview of different sections of Information Model

This breakdown is also continued within some Information Model sections, notably within the Instruction information. The Instruction information is sub-divided into four separate areas dealing with the scheduling, the configuration of the measurement parameters, the configuration of how results are reported back to one or more Collectors, and the potential suppression of measurements. The sub-division reflects that they are likely to happen on different timescales and at different times. For example, it is envisaged that the configuration of the Measurement Tasks would be fairly infrequent; updates of the Measurement Schedule could be quite common, whilst suppression would only happen when measurements need to be temporarily suspended to rapidly alleviate some unexpected problem. The breakdown of the Information Model in this manner allows the protocol implementation to transfer these pieces of information using different commands which in turn reduces the data overheads and improves the scalability of the control protocol. Figure 3 shows our initial version of a detailed breakdown of the Instruction part of the Information Model.

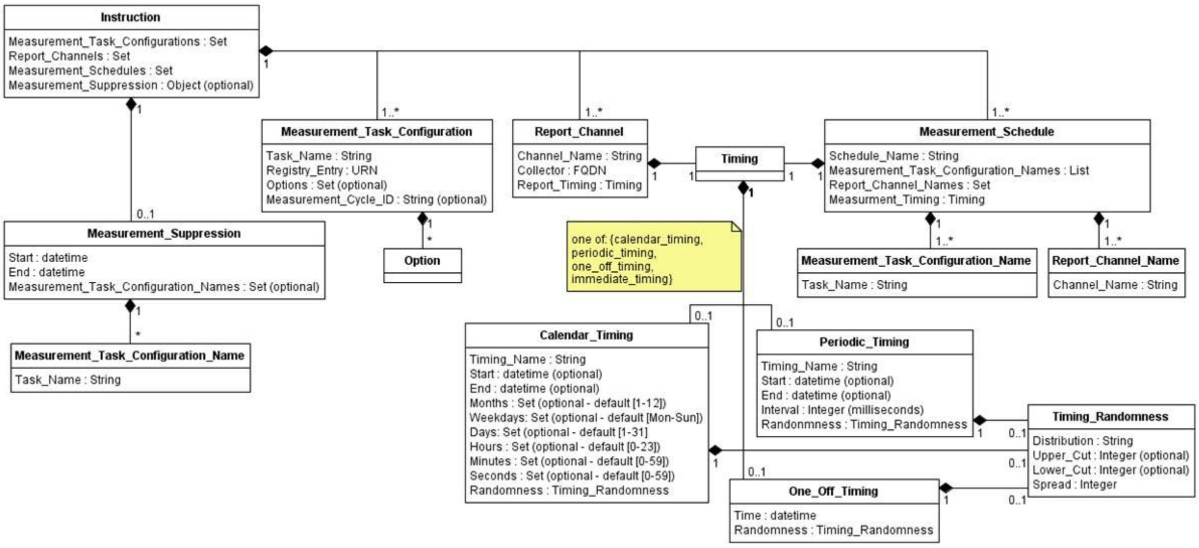

Figure 3: Details of Instruction part of the Information Model

Leone will continue to work to integrate the views of different bodies into a single IETF standard Large-Scale Measurement Information Model. The timescales are for a ‘working group draft’ in January 2014 and for working group approval in July 2014. The Information Model is on the IETF’s standards track.

While good progress has been made on some areas of the Information Model such as the Instruction and Reporting information, other sections such as Logging and Status information are still in very early discussions.

The HTTP RESTful control and reporting protocol (Section 5) is compliant with the Information Model.

5 PROTOCOL & DATA MODEL

The LMAP information model, described in the previous section, is an abstract, protocol-neutral definition of the information held and transferred to/from the Measurement Agent (MA). The information then needs to be described using a specific data model, encoded into a well-defined structured format and exchanged using a transport protocol. We have investigated several possibilities to achieve this:

- The NETCONF protocol as a control protocol and its associated YANG data model.
- The IPFIX protocol as a report protocol and its associated data model.
- The HTTP(S) protocol as both a control and report protocol and JSON data model.

Network Configuration (NETCONF) is a generic protocol to support device configuration and can be used to deliver instructions to a MA from the LMAP Controller, with the data model specified using the YANG data modelling language. Internet Protocol Flow Information Export (IPFIX) is a unidirectional, transport-independent protocol for the export of binary data records. IPFIX can be used to deliver measurement result reports from the MA to a LMAP collector. These are existing IETF protocols which potentially makes it easier to integrate them into standards management systems. However, these protocols require a number of technical issues to be overcome before they can be used in LMAP (as discussed in Leone Deliverable 3.1). As such, we are investigating an alternative new approach based on HTTP.

We propose the usage of Hypertext Transfer Protocol (HTTP) both as a control and report protocol within the LMAP framework. The semantics can be expressed in a Javascript Object Notation (JSON) encoded format. These structured representations of the information elements can then be exchanged using a REST architecture on top of the HTTP protocol. This approach has several advantages:

- HTTP is a simple protocol and easily fits within the LMAP Working Group's charter requirements.
- HTTP is widely deployed and available on potential LMAP devices (routers, smart phones etc)
- HTTP(s) port 80 and 443 are not commonly blocked by firewalls, NATs and/or other middle-boxes.
- HTTP has huge development community around it to facilitate an early prototype implementation.

The Uniform Resource Identifier (URI) design of the proposed REST Application Programming Interface (API) is described below. The high-level interaction of the MA with the LMAP Controller and Collector using this API is shown in Figure 4. Soon after deployment the MA sends a GET request to the Controller in order to retrieve its initial configuration and measurement instructions. On receiving the request the Controller verifies the identity of the MA from its UUID (universally Unique Identifier) and replies with pointers (URLs) to the other elements which contain the actual control information. Next the MA uses this received information to send a series of GET requests to retrieve this control information: the measurement set (the list of tests for the MA to run); the measurement schedule (when and with what frequency to run these tests); and the relevant reporting channel (how and when to report the measurement results). The approach of using a series of canonical GET requests, as opposed to a single GET request, allows each part to be updated independently and with their own designated frequency.

The specifications of each HTTP request and its corresponding JSON-encoded result response is described in our internet draft in more detail.

The interaction between the MA and the Controller and Collector requires some information to be pre-baked on the MA, such as the FQDN (Fully Qualified Domain Name) of the Controller and security credentials for authentication. This pre-configuration or bootstrap process is likely to depend on the access technology, for instance it could be part of the ACS initialisation (Auto Configuration Server) used in TR-069 or DOCSIS.

There are several open issues. One is whether to use the POST method for the control protocol instead of the GET method described above (GET may seem more natural but POST may be more extensible since it can carry complex information and there is no need to 'force' arguments into the strict hierarchy of URIs). Similarly, for the report protocol it may be better to use the PUT method instead of POST (as the former is idempotent). Another issue is that a Controller may need to 'push' information to a MA, so that it doesn't have to wait for the MA to initiate communication with it – for example the Controller may want the MA to make an immediate, on-demand measurement, or it may want the MA to pause (suppress) all its measurements as soon as possible. The protocols also need to deal with communication failures and be secure.

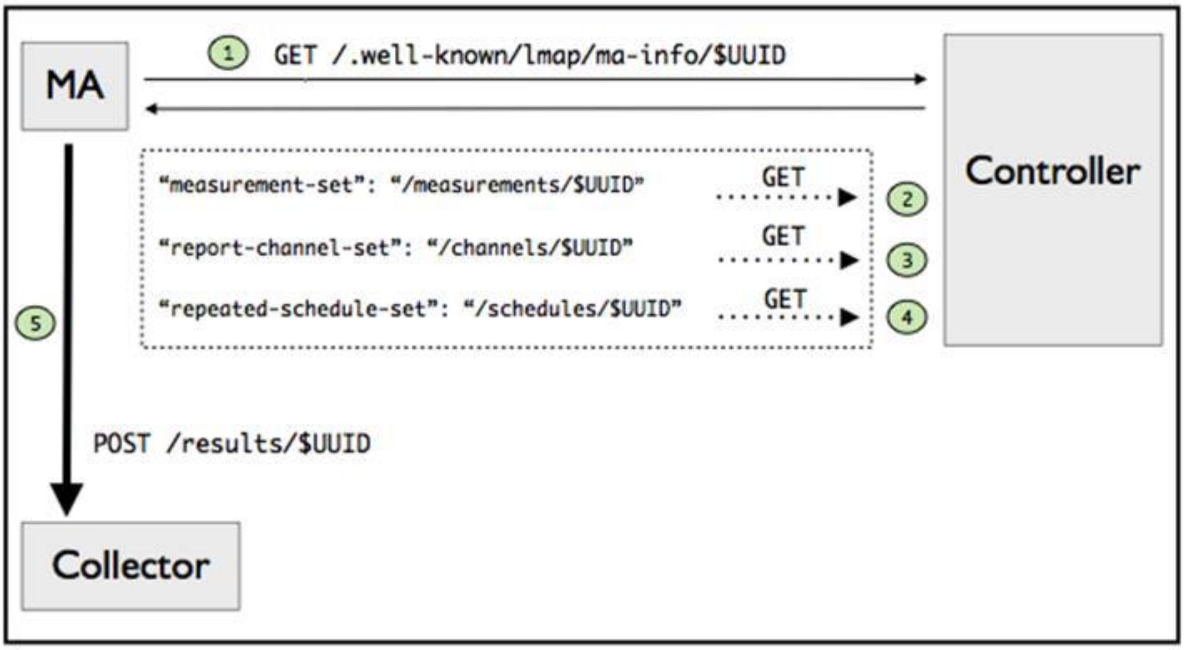

Figure 4: Interaction of the MA with the LMAP controller and collector using the HTTP protocol

6 REGISTRY & REFERENCE PATH

When the Controller sends an Instruction to a Measurement Agent it includes the Measurement Tasks that it wants the MA to execute. It is critical that the Controller and MA have the same understanding about what measurements are needed. A registry of metrics should solve this issue: a Controller simply includes a reference to the registry entry that defines the metric it wants the MA to measure; and the MA simply looks up the reference to learn what metric it should measure. Similarly, when the MA reports its measurement results, it references the registry so that the Collector can unambiguously identify the metric that was measured.

One side benefit of having a public registry of well-defined metrics (and the methods to measure them) is that measurement results are comparable even if they are performed by different implementations, in different networks and even using different control or reporting protocols (for example, one implementation might use an http-based protocol, as described in the previous section, whilst another might use a Broadband Forum based protocol or even a proprietary one).

Another benefit is that the registry could serve as an inventory of useful and used tests that are normally supported by different implementations of MAs. We believe that the registry should only contain a few tightly-defined metrics, so that they have only a few open parameters which don't affect the nature of the tests (such as source and destination address). This learns from the failure of the previous attempt to define a registry (by the IETF's IPPM working group); the problem was that the metric definition left too many degrees of freedom for the actual implementation – for example it didn't define whether the packets were TCP, UDP, ICMP or something else.

As well as the metric itself (perhaps "UDP packet latency"), the registry also needs to define auxiliary items ("sub-registries") such as the scheduling strategy (perhaps periodic or Poisson scheduling, or a singleton test); the output type (perhaps the raw measurements or their mean); and the environment (perhaps only make the measurements when there is no cross-traffic). We have explored two different approaches for structuring the registry. The issue is whether the sub-registries are completely independent, with the Controller free to choose any combination of entries, or whether the sub-registries are hierarchical so that a single registry entry defines a choice about each of the sub-registries. The latter approach, after discussion at the IETF, seems preferable; the first creates too much work for implementers, as they have to check their implementation works for every single possible combination, many of which may never actually be used. On the downside, there is a bit more standardisation work defining and maintaining the registry.

We have also working at the IETF to create a registry of measurement points and path components. The motivation is to provide an unambiguous way to describe the scope of the path over which a measurement is made, since general terms like "end-to-end" are open to several interpretations (What is an end? Is the home network included?). This could be useful both for diagnosis (where the same metric may be measurement over several different path scopes) and for comparison (where the same metric is measured on different network infrastructures).

Some of the reference points are relatively obvious: subscriber device, access service demarcation point, intra IP access and globally routable address gateway. Perhaps a less obvious reference point is the resource transition point, which marks the point of transition from dedicated to shared components, ie. from dedicated resources serving an individual subscriber to common resources shared by multiple subscribers.

7 SUBSCRIBER PARAMETER DATABASE

The Subscriber Parameter Database (SPD) is the framework function responsible for supplying network and subscriber information to the Controller and data analysis tools. Such data could include:

- Subscriber information such as product, usage caps, traffic management policy and the subscriber's timezone
- Network information such as access technology, line length, equipment type, exchange id and geo-location (especially for mobiles)
- Network status information such as a DSL modem's actual rate, line errors, interleaving and network utilisation

Such information is often critical to work out the right set of measurements for the Measurement Agents to perform – a test shouldn't overwhelm the typical capacity of the line, for instance. The information is also important for analysis of the measurement data. For example a regulator may want to compare the measured speed with the rate in the subscriber's broadband contract, whilst an ISP may need to know the subscriber's modem type, local aggregation node and exchange, in order to determine which other subscribers may be affected by a fault.

Some subscriber information is naturally reported by the MA, either because it knows it directly or it is transmitted to the MA by using TR-069 (or DOCSIS equivalent). Examples include the sync speed where the MA is a home gateway, the type of modem, power saving parameters and some DSLAM or channel configuration parameters. We currently assume that all MA context information is included in every Report to avoid the complexity of having to implement policy languages to control its dissemination. Such parameters may be included as additional columns in the result rows or in the header part of the report for more generic context. It is included in the Information Model.

Other information is not naturally known by the MA. An issue is whether it should be transmitted directly from the SPD to the data analysis tools, or whether it should be transferred via the MA (thus SPD to MA to Collector to data analysis tools). The circuitous route has been suggested to ensure the information is up-to-date (people change broadband contracts and probes get given to other people). However the data ultimately comes from network and account OSS that may not be fully up to date at the time the MA performs its measurement. It is not realistic for this vast wealth of subscriber parameter data to be transmitted to the MA and used to enhance every measurement. Besides the overhead, another issue is that the ISP would have to take great care to ensure that the MA only gets the information about the correct subscriber (even if the MA was moved to another household). Similarly an MA reporting to different Collectors may have to carefully select which line/subscriber data was sent to each.

A particular scenario that needs further analysis is where an ISP-run measurement system reports results to a third party such as a regulator. In some jurisdictions it has been claimed that data privacy considerations may be easier if only the MA (and not the ISP) sends subscriber information.

Finally an open issue is whether the interfaces with the SPD should be standardised. At least some aspects can be considered proprietary or at least specific to the implementation within the network OSS, for example an SQL database interface or Hadoop big data repository or an exchange of CSV files, as SamKnows does with some ISPs.

8 CONCLUSIONS

The main purpose of architecture work is to guide the development of the detailed technical solutions. So we have studied the motivations for an ISP to measure network performance and the weaknesses of current approaches, and thus derived the goals for our work and the simplifying assumptions we should impose on solutions. We believe that the main goal is for large-scale, flexible and standards-based solutions, whilst perhaps the most important constraint is the assumption that the measurement system is under the direction of a single organisation responsible for ensuring that it has no adverse impact on security, data privacy or user experience.

The architecture itself contains four main functional components: Measurement Agents and Measurement Peers which jointly generate test traffic and measurement some metric of interest associated with its transfer (such as 'time to transfer a test file' or 'UDP packet loss'); a Controller which manages the Measurement Agents (MAs); and a Collector to gather the measurement results from the MAs.

We are researching new measurement tests (reported in another Leone document) and we are pursuing the creation of a registry (by the IETF) of the most useful and used metrics. We are researching the interfaces between the Controller and MA, and the MA and Collector, and have started working with colleagues at the Broadband Forum and IETF to standardise them, in particular by initiating the creation of a new IETF working group called LMAP.

There are several open issues, for example how to make an 'admission control' check that the test traffic associated with making a measurement won't overload the end user, measurement peer or network, and how to enhance the measurement results with information about the subscriber, such as the Mbit/s rate in their broadband contract.

We have made a first proposal for a protocol implementing the MA interfaces, which is based on HTTP with information encoded in JSON. The approach is motivated by their wide deployment. In order to reach agreement about what information needs to be transferred over the protocol, we are also developing an abstract, protocol-neutral definition of the information; we hope that such an information model could become the single universally-accepted standard, as this would allow some degree of interoperability between our HTTP-based solution and other solutions, for instance perhaps based on the Broadband Forum protocol TR-069.

Future work will study the various open issues and develop the information model, data model and protocol, and the registry.

Other important aspects covered in other Leone documents include: privacy and security (since a compromised measurement system could be used to launch DDoS attacks for instance); how the measurement information is analysed, visualised and integrated with existing network management systems; and our plans for implementing, testing and trialling our proposals. These are also subject to on-going research and will all help to validate and refine our architecture.

REFERENCES

[1] Large-Scale Broadband Measurement Use Cases; draft-linsner-lmap-use-cases; Marc Linsner, Philip Eardley, Trevor Burbridge

[2] A framework for large-scale measurements; draft-eardley-lmap-framework; Philip Eardley, Trevor Burbridge, Al Morton

[3] A Framework for Large-scale measurements; Future Network & Mobile Summit 2013, July 3rd -5th, 2013, Lisbon, Portugal; Marcelo Bagnulo, Philip Eardley, Trevor Burbridge, Sam Crawford, Juergen Schoenwaelder

[4] Information Model for Large-Scale Measurement Platforms (LMAP); draft-burbridge-lmap-information-model; Trevor Burbridge, Philip Eardley, Marcelo Bagnulo, Juergen Schoenwaelder

[5] Large MeAsurement Platform Protocol; draft-bagnulo-lmap-http; Marcelo Bagnulo, Trevor Burbridge, Sam Crawford, Juergen Schoenwaelder, Vaibhav Bajpai.

[6] A registry for commonly used metrics; draft-bagnulo-ippm-new-registry; Marcelo Bagnulo, Trevor Burbridge, Sam Crawford, Philip Eardley, Al Morton

[7] A registry for commonly used metrics. Independent registries; draft-bagnulo-ippm-new-registry-independent; Marcelo Bagnulo, Trevor Burbridge, Sam Crawford, Philip Eardley, Al Morton.

[8] A Reference Path and Measurement Points for LMAP; draft-morton-ippm-lmap-path; Marcelo Bagnulo, Trevor Burbridge, Sam Crawford, Philip Eardley, Al Morton.

The references in turn contain numerous references with further information. Prior measurement systems include:

[9] BISMark: projectbismark.net/

[10] Figaro: www.ict-figaro.eu/

[11] MLabs: www.measurementlab.netOokla: http://www.ookla.com/

[12] Netalyzr: http://netalyzr.icsi.berkeley.edu/

[13] perfSonar: www.perfsonar.net/

[14] RIPE Atlas: labs.ripe.net/atlas

[15] SamKnows: http://www.samknows.com/broadband/methodology

[16] I. Canadi, P. Barford, J. Sommers, 2012. Revisiting broadband performance. In Proceedings of the 2012 ACM conference on Internet measurement conference, DOI=http://doi.acm.org/10.1145/2398776.2398805