A Framework for Large-scale measurements

Marcelo BAGNULO[1], Trevor BURBRIDGE[2], Sam CRAWFORD[3], Philip EARDLEY[2], Juergen SCHOENWAELDER[4]

[1] *Universidad Carlos III de Madrid, Spain, marcelo@it.uc3m.es*

[2] *BT, UK, trevor.burbridge@bt.com and philip.eardley@bt.com*

[3] *SamKnows, UK, sam@samknows.com*

[4] *Jacobs University Bremen, Germany, j.schoenwaelder@jacobs-university.de*

Abstract: Although there are many "measurements" of the Internet today, they operate independently. So measurement results are hard to compare and systems are hard to scale and integrate into existing network management. This paper presents an architecture for large-scale measurements, analyses the interactions between the various components and discusses how to standardise a couple of the interactions. Such a standardised architecture could allow measurement capabilities to be embedded into every home gateway and edge device, thereby enabling (for example) operators to provision and manage their networks more effectively.

Keywords: measurement agent, controller, collector, test registry, data model.

1. Introduction

There are many "measurements" of the Internet today. To give some examples: The routers, switches and other equipment in a network report their utilisation and status to an operator's management system. Large businesses, CDNs and over-the-top providers measure the end-to-end performance of their applications in order to compare their network providers, assess the impact of a new application and improve their caching and WAN optimisation. End users may run a web-based speed test to check the performance of their broadband provider. Regulators use probes deployed in some consumer homes to benchmark ISPs. Operating systems may test the impact of a new protocol such as IPv6 before deciding whether to use it.

The story is a wide diversity of independent measurement techniques each fulfilling a narrow need and can be seen as another instance of "the Internet only just works" [1] – each technique developed, deployed and operating independently. We believe that a framework for large-scale measurements enables a more cohesive approach, so that the capability is more pervasive and manageable and performance metrics can be directly compared.

In this paper we outline the Leone framework (or architecture) for measurements. The requirements are that it elegantly handles:

- Scale: our vision is that a measurement capability is embedded in every home gateway and edge device, such as set-top-boxes and tablet computers

- Diversity: the measurement function may be in a specific hardware probe, embedded in another device or an application running on a general purpose device
- Extensibility: new tests may be needed, for example an improved test methodology or to measure a performance metric not previously considered important (eg bufferbloat)
- Consistency: so that it is meaningful to compare measurements made of the same metric at different times and places
- Multi-dimensional information: with ubiquitous measurement functions, then it should be possible to combine information from several viewpoints in order to identify where a network fault is
- Privacy and security: the measurement capability must not open a new attack vector, nor reveal private information about users or networks
- Quality of experience: measurements should help improve the quality experienced by end users and not focus only on network parameters.

The contributions of this paper are to:

- Define the Leone framework for large-scale measurements (Section 2)
- Analyse interactions between the various elements (Section 3)
- Describe a way of standardising some of those interactions, namely a registry of tests (Section 4) and data models for the interaction of a measurement agent with its Controller and Collector (Section 5).

2. Outline of the Leone framework

The Leone framework for large-scale measurements has four basic elements: Measurement Agents, Test Servers, Controllers and Collectors – as well as two optional components: Initialiser and Subscriber Parameter Database. In addition there are the Data Analysis Tools that use the measurement information, which in turn assist the operator's OAM (Operational and Maintenance) capability.

A **Measurement Agent** (MA) and **Test Server** (TS) jointly perform an active measurement test, by generating test traffic and measuring some metric associated with its transfer over the path from the TS to MA (or MA to TS); for example the time taken to transfer a 'test file'. A MA may also conduct passive testing through the observation of traffic (ie without the involvement of a TS); for example about an end user's mix of applications. The MA and TS functions are implemented either in specialised hardware or as code on general purpose devices like a PC, tablet or smartphone. Section 4 considers a registry for tests.

The **Controller** manages a MA by instructing it which tests to perform and when. For example it may instruct a MA at a home gateway: "Run the 'download speed test' with the test server 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". The Controller also manages a MA by instructing it how to report the test results, for example: "Report results once a day in a batch at 4am; if the end user is active then delay the report 5 minutes". We refer to these as the **Test Schedule** and **Report Schedule**. A Controller can also initiate one-off tests, as well as regular ones.

The **Collector** accepts a Report from a MA with the results from its tests. It may also do some limited processing on the results – for instance to eliminate outliers, as they can severely impact the aggregated results.

Therefore the MA is a Leone-specific device that initiates the test, gets instructions from the Controller and reports to the Collector. On the other hand, the TS interacts only with the MA and indeed doesn't have to be Leone-specific - for example the MA could time how long it takes to get the Google home page, with the Google server acting as the TS.

There are also two components that we are discussing including in the Leone framework:

An **Initialiser** configures a MA with details about its Controller, including authentication credentials. Possible protocols are SNMP, NETCONF or (for Home Gateways) CPE WAN Management Protocol (CWMP) from the Auto Configuration Server (ACS) (as specified in TR-069 [2]).

A **Subscriber Parameter Database** contains information about the line, for example the customer's broadband contract (2, 40 or 80Mb/s) and the line technology (DSL or fibre), or even the time zone the MA is located, which may affect the choice of Test Schedule. For example, a download test suitable for a line with an 80Mb/s contract may overwhelm a 2Mb/s line. Other factors could be the line technology, line performance, type of home gateway or version of the MA itself. Another example is if the Controller wants to run a one-off test to diagnose a fault, then it should understand what problem the customer is experiencing and what tests have already been run.

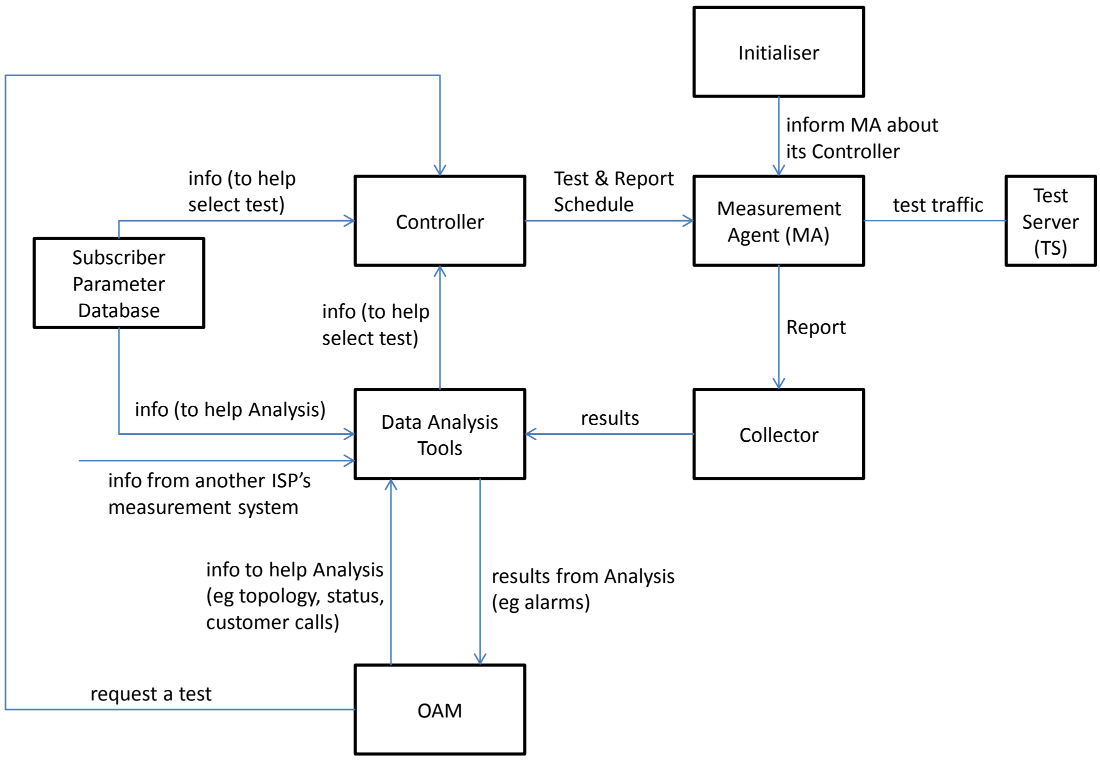

Figure 1: Leone Architecture for large-scale measurements

The **Data Analysis Tools** receive the results from the Collector. For a regulator it might be sufficient to simply aggregate the collected data and to provide a condensed report of what has been observed. But for ISPs much more elaborate data analysis is required. Measurement data needs to be correlated with topological information (at various layers) and status information in order to isolate the faulty part of the network; this requires interfacing more tightly with the operator's **OAM**, in order to obtain the information needed for a more detailed data analysis. Within Leone we are researching and developing tools to perform data integration and root cause analysis and visualise measurement data and analysis results, with the objective of enhancing an operator's OAM, so that it can better understand and manage its network and service performance and troubleshoot problems.

Figure 1 sketches the interactions between the various components, which the next Section discusses further.

3. Interactions between the components

In this section we consider the relationships and interactions between the various components of the Leone framework.

3.1 Test Server and Measurement Agent

Each MA can test against multiple TSs, and each TS measures against many MAs. For example an MA on a Home Gateway may use Test Servers at the BRAS (Broadband Remote Access Server), at the Internet Peering Point of an ISP or even in another network, in order to isolate any problems or performance bottlenecks. It is also possible for a Test Server to be incorporated into the Home Gateway in order to allow testing of the home network's performance, by installing MAs on other in-home devices.

3.2 Measurement Agent acts autonomously

In the Leone framework, once the MA gets its Test Schedule from its Controller then it acts autonomously, in terms of operation of the tests and reporting of the result.

Firstly, this means that the MA initiates measurement tests. For the typical case where the MA is on a home gateway or edge device, this means that the MA initiates (for example) a 'download speed test' by asking the TS to send the file. In the case the test should be performed when there is no user traffic on the link, the MA knows whether the end user is active and therefore whether to start the test or delay it. Also, if the MA is behind a NAT then the TS naturally learns its public-facing IP address, and also having the Schedule on the MA avoids it having to check frequently with the Controller.

Secondly, all measurement results are sent from the MA – even for an 'upload test', where the TS therefore needs to report the result back to the MA, which reports the result in turn. The reason is that it is easier to secure the reporting process, for example with a unique certificate for each MA-Collector pair, so that the Collector is confident the results really do originate from the MA.

3.3 Controller and Measurement Agent

In the Leone framework each MA may only have 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 Schedules. This simplifies the design of MAs (critical for a large-scale infrastructure) and allows a Test Schedule to be tested on specific types of MA before deployment to ensure that the home user experience is not impacted (due to CPU, memory or broadband-product constraints). However, an operator is likely to have several Controllers, perhaps with a Controller for different types of MA (home gateways, tablets) or locations (Ipswich, Edinburgh).

To avoid problems with NATs and firewalls, the MA 'pulls' its configuration from its Controller (as identified by the Initialiser).

Open issues:

At the moment it is an open issue whether there is any negotiation between a Controller and its MA, or whether the Controller simply instructs the MA by sending its Test and Report Schedules.

The argument for negotiation is that occasionally the MA may be updated with enhanced versions of existing tests. It is easier for the Controller to learn the MA's capabilities directly from the MA than from a management system and it avoids any mis-synchronisation. On the other hand, this makes the Controller-MA protocol more complicated, increases the MA's resource requirements and increases the complexity of the Controller when it decides how to schedule tests across numerous MAs or when it deploys a new Test Schedule to potentially millions of MAs. At the moment we believe that the second, 'no negotiation' approach is better, since the management system knows the MA's capabilities. However, occasionally there may be an error, so it may be useful to have an additional function that allows the Controller to ask the MA to confirm its capabilities.

3.4 Interactions of Subscriber Parameter Database

Certain tests benefit from knowledge of the subscriber's broadband contract, line technology, line performance, home gateway and so on. The Controller needs to obtain that knowledge so that it can adapt the Test Schedule appropriately.

Open issues:

At the moment it is an open issue whether the Subscriber Parameter Database only interacts with the Controller (and where necessary the MA uses an existing protocol to report its parameters to the OSS, such as TR-069 [2] for home gateways) or whether the MA learns information directly from the Subscriber Parameter Database and then informs the Controller.

The argument in favour of the latter (MA-Database) approach is that that it enables the test result data to be enhanced immediately (as it is produced), which avoids the more problematic operation of attempting to look up synchronous historical values for line/account parameters at a later date, especially as some of these historical parameters may not be stored in existing systems. However, this requires the OSS to keep the Subscriber Parameter Database updated in real time, which is also problematic. Further, it increases the complexity, overhead and performance requirements on the MA: the MA needs to continually retrieve data from the Database, in sync with its scheduled tests, but with extremely careful security (since line account information is transferred and the MA may be on an untrusted platform).

Therefore at the moment we favour the approach of 'no interaction of the MA and Subscriber Parameter Database'.

3.5 Collector and Measurement Agent

In the Leone framework there is no negotiation between the Collector and MA. The measurement results are simply delivered. This could occur periodically (e.g. every hour) or based upon the volume of data collected (e.g. when data exceeds 1MB) or some combination thereof. The result of a one-off test needs to be reported immediately. If the Collector is unavailable, a strategy needs to be specified – this might be to retry, failover to another server, discard oldest results or to stop testing.

The Report includes details about each test, such as the type of test, the TS used and the time of the measurement. This avoids the Collector having to get this information from the Controller, which would be less reliable. The Report also details any abnormal conditions that may be result-impacting, such as shortage of CPU for the MA, significant user traffic during the measurement, or even if the test had to be abandoned. It is anyway impossible for the Controller to know about such conditions.

3.6 Interactions of Data Analysis Tools and OAM

The OAM system feeds topological information and status information into the Data Analysis Tools and the Tools feed data analysis results into OAM systems (eg by creating events or alarms in the OAM systems). It may also be possible for some interaction between a Controller and the Data Analysis Tools or OAM. For example analysis of the results from one test could trigger a request for the Controller to initiate another specific test, perhaps to help narrow down the cause of a problem.

The OAM system may want to trigger tests on demand. For example, when a customer rings up with a problem, the helpdesk would like to run immediate tests to try and identify and resolve the issue before the call ends. In such a case the OAM system interacts with the Controller.

The Data Analysis Tools may get "multi-dimensional" information from many sources beyond the Collectors, for example BGP reports, topology information, OSS statistics and so on, as well as potentially (suitably anonymised) results from Collectors in another measurement system.

3.7 Measurement system and outside parties

In the Leone framework the measurement system is under the direction of a single organisation. This constraint ensures clear responsibility for both the data and the user experience, which are critical given that a misbehaving large-scale measurement system could potentially harm user privacy and act as a botnet.

However, Leone also has the vision that data from multiple organisations can be combined in order to really understand network and service behaviour. For example information from another operator could help identify that a fault is on a transit network that they share.

<u>Open issues</u>

It is an open issue exactly the form of interaction with an outside party. For some types of interaction, for example with a regulator, it may be entirely through off-line, person-to-person negotiations – a monthly meeting where the latest batch of results are presented on disc and the next month's tests are discussed. A similar example is where measurement results are archived (or 'curated') for the benefit of researchers.

Where the measurement information is being used to help identify faults, then a more dynamic system may be needed. The outside party makes a request and receives results, with both interactions mediated by policy management to ensure that its requests are allowed and to make sure that the results are obfuscated an appropriate amount; all interactions could go through a single policy enforcement point. Critically the operator of the measurement system is responsible for the final Test Schedule, so it must work out how to smoothly incorporate the outside party's requests into the Schedule.

It is an open research question to what degree the raw test results should be obfuscated. The question is not purely about protection of privacy; for example, it may be important to ascertain whether an MA operated in a set top box by a service provider is in the same home network as an MA measuring broadband line performance on behalf of an ISP – this example suggests that we may need a secure way to refer to network location without compromising the privacy of a user or confidentiality of any organisation.

4. Test registry

One fundamental piece for the Leone architecture is a registry for the tests used in the platform. In order to have such a registry, we need to create a number of namespaces whose values will be recorded by the registry and will uniquely and precisely identify the tests. We propose a first version of such a registry in [3], potentially for standardisation by the IETF's IPPM working group.

The motivation for having such registry is to allow a Controller to request a Measurement Agent to execute a measurement using a specific metric. Such requests can be performed using any control protocol that refers to the value assigned to the specific metric in the registry. Similarly, the MA can report the results of the measurement and by referring to the metric value it can unequivocally identify the metric that the results correspond to.

Having a public registry for tests has a couple of side benefits. First the registry could serve as an inventory of useful and used tests that are normally supported by different implementations of MAs. Second, the results of the metrics would be comparable even if they are performed by different implementations and in different networks, because the metric is properly defined.

This would simplify large-scale deployments as different implementations would execute the same test upon a request for the same registry value.

5. Data model structure and language

The interfaces between the Controller and MA and between the MA and Collector need to be well defined in order to enable interoperability on a basic set of functions. Furthermore, the interfaces should be flexible enough to deal with extensions, either additional standards extending the basic interface or proprietary extensions enabling the development of new functions that may be considered later for standardisation.

<u>Open issues:</u>

The Controller configures the MA with Test and Report Schedules and it will provide additional test parameters (eg the set of TSs to run tests against). Since this is primarily a configuration management problem, existing configuration management standards such as NETCONF and YANG may be used. YANG is a data modelling language originally designed to support the data modelling needs of the network configuration protocol NETCONF. However, YANG can also be mapped to other protocols; indeed in [4] we proposed to map YANG data models to RESTful APIs. A RESTful API likely provides a convenient interface between the MA and the Controller. However, at the time of writing, it is not clear whether the IETF will standardise the RESTful API for YANG. Note that for this particular use case, it may be required to decouple the protocol session initiation from the request/response interaction during the session since MAs may likely be located behind NATs. In [5] we sketch a data model for large-scale measurements using YANG. It may be better to integrate the Test and Report Schedules into a single Schedule.

The MA reports results to the Collector. To achieve this, it may be possible to use the IPFIX protocol, originally designed to transfer network flow information from a router to a collector, as we propose in [6]. The core IPFIX protocol assumes a flat data model and provides a flexible and efficient encoding, in particular for repetitive data. It needs to be investigated whether IPFIX and its data modelling framework provides the required mechanisms to support MAs efficiently. Again a RESTful API may be more convenient.

At the IETF, the proposed LMAP working group [7] could standardise a data model for test control and reporting. LMAP could also standardise the Controller-MA and MA-Collector protocol(s), which transfer information using the data model.

6. Comparison of Leone framework with existing measurement systems

There are relatively few systems designed to perform controlled measurement across a large number of network points. We look at a few of the architectures that perform hardware-based measurements in the sub-sections below. We have omitted software-based solutions for brevity, although it should be clear that software based tests should also conform to the framework outlined in this paper. With most software-based measurements (such as Ookla [8]), the controller function is performed by the end user, who requests that the test is conducted. The results from the test are sent to the user display in addition to a collector in the network. Care should be taken when comparing results from software-based measurement agents since the panel, test schedule, platform and local cross-traffic are uncontrolled, even though the same test may be run between similar end points.

6.1 Broadband Forum

The Broadband Forum [9] (previously the DSL Forum) defines a framework for the management of CPE including Home Gateways and other devices such as Set Top Boxes through a number of technical documents. TR-069 [2] defines how a CPE is managed by a single ACS (Auto Configuration Server) and how the device communicates to the ACS using CWMP (CPE WAN Management Protocol). In spirit this is similar to the Leone MA communicating with the Initialiser and Controller, although CWMP was never envisioned for the communication of complex test schedules or the routine reporting of test results. However, for an MA implemented on a home gateway (or other TR-069 compliant device) the ACS would fulfil the role of the Initialiser and configure the Controller's location (through a yet-to-be-defined data model). Of note amongst the set of other Broadband Forum documents is TR-143 which defines a data model for TR-069 devices that allows throughput and delay testing. The testing capabilities are currently limited to an on-demand upload/download throughput test and the configuration of the device to respond to network ping tests in order to measure network latency. Using the ACS to manage a test schedule and to continuously demand single tests across large numbers of devices is not scalable. Performing tests from the network towards the CPE will be hampered by NATs and firewalls, where the MA is not implemented on the Home Gateway. Furthermore there is no method to ensure that tests are only run in the absence of user traffic. For these reasons the Leone participants are currently working within WT-304 of the Broadband Forum to extend the current capabilities.

6.2 SamKnows

SamKnows [10] operates panel-based measurements using hardware whiteboxes on behalf of a number of ISPs and regulators. The approach taken is very similar to the Leone target architecture (and indeed the Leone testbed is implemented with SamKnows) with MAs being given test schedules by a Controller and reporting to a central Collector. The Leone target architecture extends the current SamKnows capabilities by assuming that MAs can be embedded in multiple devices, and that Controllers use a single standard way of controlling the tests and collecting results. The Leone architecture also addresses one of the main implementation barriers of using SamKnows which is the accurate enhancement of measurement data with device, line or product information through the definition of the Subscriber Parameter Database. Since Leone envisages a more extensive and varied deployment of MAs than in the current SamKnows deployments, critical extensions are to authenticate the line the MA is using and to define the network path that is being measured.

6.3 Bismark

Bismark [11] is an academic measurement platform using hardware measurement probes in customer premises similar to the SamKnows architecture. The probes run similar active tests on a per-user schedule and report back to a central server along with capturing several passive measurements such as application throughput and usage. The inclusion of passive measurements means that implications for user privacy are greater, although like most measurement platforms these are performed under user agreement and held according to data protection regulation. These concerns also drive the Leone architecture away from allowing multiple Controllers, enabling the MA to have a single end-user agreement for the testing that takes place including the purpose of the data collection.

6.4 Atlas

Atlas [12] is a measurement framework operated by RIPE using small form factor USB-powered active probes. Due to the low capabilities of these probes tests tend to be around connectivity or capability testing of the network. The user allows the probe a maximum amount of user activity over which RIPE will operate their own tests. Atlas is a single monolithic measurement platform. But uniquely it allows users to define and execute their own tests on other user’s probes (subject to careful control and agreement). This works for Atlas since (a) user-defined measurements are deployed on a limited number of probes; (b) the bandwidth used is limited; (c) a probe has no access to user traffic; (d) a user has agreed to the measurement probe being placed in their own network and is aware of the arrangements for other users. On the other hand, Leone assumes that MAs will be deployed on devices that do have access to user traffic and that can use the full capabilities of the network when required. In addition, the heterogeneous nature of the MAs means that the test schedule must be carefully managed and tested for different types of device to ensure that the device and line are capable of operating the test schedule successfully. These reasons, combined with the issues of data protection discussed previously, push Leone towards having a single Controller responsible for any particular MA.

7. Conclusions

This paper presents an initial framework for large-scale measurements. A widely accepted framework for large-scale measurements, with its critical aspects standardised, would enable a pervasive, manageable and comparable capability that would help operators provide better broadband for end users. We have started working towards the standardisation of various aspects, particularly a registry for test definitions and an information model for the Schedules (Controller to Measurement Agent) and the Report (Measurement Agent to Collector).

We conclude that each Measurement Agent should only have a single Controller at any one point in time and a Measurement Agent should act (semi-) autonomously. Also, a measurement system should be under the direction of a single organisation. We also tentatively conclude that the Measurement Agent's interaction with its Controller and Collector should be without negotiation; in other words the Controller simply instructs a Measurement Agent with a Test and Report Schedules, and the Measurement Agent simply delivers the Report (with the results of the measurements) to a Collector. Further, a Measurement Agent should not directly interact with the Subscriber Parameter Database.

We conclude that the framework offers great promise to meet the requirements listed in the Introduction. A standardised framework, with the simplifying constraints outlined in the previous paragraph, can **scale** and allow a **diversity** of types of Measurement Agent. The "single organisation" assumption makes the **privacy and security** requirements easier to meet and clarifies who is responsible for them. The registry of tests ensures **consistency** (results are comparable) and **extensibility** (just add new tests). The Data Analysis Tools analyse **multi-dimensional information** – including results from measurements beyond the local network as well as other information like topology – which enhances the operator's Operational And Maintenance capability, by helping it to isolate where a network fault is, optimise investment in new capacity and understand the likely impact of potential new services and new features.

8. References

[1] Why the Internet only just works. M Handley. BT Technology Journal Vol 24 No 3 July 2006

[2] TR-069, CPE WAN Management Protocol, from [9]

[3] A registry for commonly used metrics. Independent registries. M. Bagnulo, T. Burbridge, S. Crawford, P. Eardley, A. Morton. January 2013 draft-bagnulo-ippm-new-registry-independent

[4] YANG-API Protocol. Bierman, A., M. Bjorklund. November 2012. draft-bierman-netconf-yang-api

[5] A YANG Data Model for Large Scale Measurements. J. Schoenwaelder. February 2013 (to appear). draft-schoenw-lmap-yang

[6] An LMAP application for IPFIX. M. Bagnulo, B. Trammell. February 2013. draft-bagnulo-lmap-ipfix

[7] LMAP BoF, Large-Scale Measurement of Broadband Performance (LMAP), http://trac.tools.ietf.org/bof/trac/#TimeframeIETF86Orlando

[8] http://www.ookla.com/

[9] Broadband Forum. Technical Reports (TRs) and information about work in progress (WT) from http://www.broadband-forum.org/

[10] SamKnows. White papers at http://www.samknows.com/broadband/methodology

[11] The BISMark Project: Broadband Measurements from the Network Gateway. N. Feamster. May 2011. projectbismark.net/

[12] RIPE Atlas, First results. R. Kisteleki, May 2011. labs.ripe.net/atlas