

lmap
Internet-Draft
Intended status: Informational
Expires: April 21, 2016

J. Brzozowski
Comcast Cable
V. Gehlot
S. Kulkarni
Villanova University
October 19, 2015

Software Architecture and Guidelines for Creating an LMAP Reference
Implementation
draft-jjmb-lmap-reference-implementation-guide-00

Abstract

This document describes a software architecture and reference implementation guidelines for the framework for Large-Scale Measurement of Broadband Performance (LMAP). The IETF-LMAP working group proposed a draft that details a logical architecture for performance measurements in broadband networks on a large scale. The IETF-LMAP proposal contains the operational concepts for such a platform. However, since the standardization is in the initial stages with no official existing implementation, the proposal leaves many details of the specification up to the implementer to discover.

Status of This Memo

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

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

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 21, 2016.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

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

Table of Contents

1. Introduction	2
2. Background	3
3. Implementation Architecture	3
4. Implementation Details	4
4.1. Probes	4
4.2. Configuration Management Servers	5
4.3. Plugin Servers	5
4.4. Update Servers	5
4.5. Probe Cache Servers	5
4.6. Refined Data Servers	6
5. Relationship to LMAP	6
6. IANA Considerations	6
7. Security Considerations	6
8. Acknowledgements	6
9. Informative References	7
Authors' Addresses	7

1. Introduction

Network performance and its related quality of service (QoS) have a significant impact on packet latencies and packet drop rates. High packet latencies and drop rates adversely affect loss-sensitive and delay-sensitive applications such as online games and real-time packetized voice and video. In addition, the latency imposed by Domain Name Servers (DNS) [RFC1034] for host name lookups also further degrades a customers web browsing experience. Network path variability between the consumer and destinations on the Internet also play a significant role in overall performance and customer experience. Finally, inadequate or an unstable downstream bandwidth can adversely affect the quality of streaming video, an increasingly popular resource-heavy service. Therefore, it is essential that the end to end capacity and performance fall within a quantifiably "acceptable" range during most times of expected network usage.

Unfortunately, the QoS and quality of experience (QoE) can be quite fluid and extremely dependent on the time of day/night and on planned or unplanned network events across wide geographical areas. If the QoE falls outside of the "acceptable" range, the customer is likely

to perceive the service provided by the ISP as being deficient. This adverse perception results in either a telephone call or some form of an inquiry to the ISP's customer service department thereby adding to the business cost of operation.

Thus, direct measurement of every customer's network performance metrics such as latency, latency jitter, packet drop ratio, path, and upstream and downstream bandwidth is essential. Since the number of customers of an ISP can be in the millions, the architecture of such a real-time data collection and analysis system must be both scalable and extensible. Therefore, we need an appropriate framework within which to design and deploy such a large-scale and (potentially) geographically diverse solution. Such considerations are the motivation behind our work.

2. Background

The Internet Engineering Task Force (IETF) has a working group that is drafting standards for Large-Scale Measurement of Broadband Performance (LMAP) [RFC7594]. The IETF-LMAP group's efforts on standardization of performance measurements span home and enterprise edge routers, personal computers, mobile devices and set top boxes. The work described in this paper started in 2012, and precedes the formation of the IETF-LMAP working group by almost a year. Our work complements the LMAP framework by adding crucial implementation details to its three components that are largely incomplete and thereby advances the IETF's standardization effort in this area. Although the work that we present in this paper precedes the IETF-LMAP proposal, our work nevertheless fits neatly within this framework and expands the discovery process so as to aid in the creation of functional models, performance models, and simulation-based architectural experimentation.

3. Implementation Architecture

The Network Measurement and Monitoring Architecture (NMMA) that was designed and implemented by our team consists of six major components (or modules) [Kulkarni2016]. These six modules are: the Probe, the Configuration Manager, the Probe Cache Server, the Update Sever, the Plugins Server and the Refined Data Server. Figure 1 below gives the schematic.

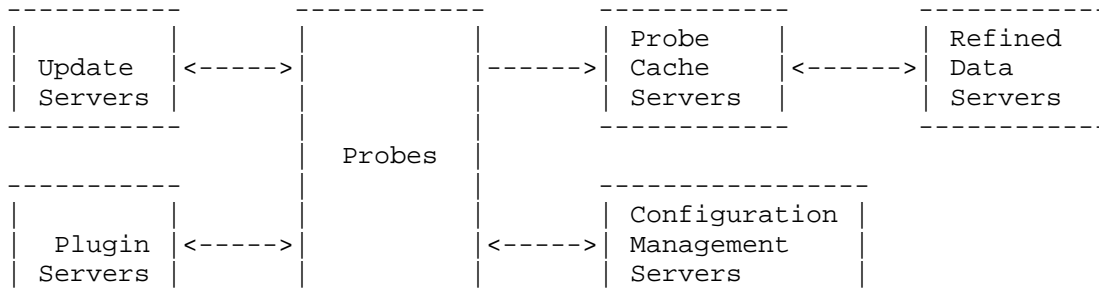


Figure 1: Software implementation architecture for network measurement and monitoring

A desirable property of a network measurement and monitoring system is that it be extensible and scalable. For example, future tests should be easily integrable into existing framework and any replication needed to handle increase in volume and velocity of measurement data should be (semi) automatic.

4. Implementation Details

This section elaborates on the requirements imposed on each of the software component that makes up the measurement and monitoring system depicted in Figure 1.

4.1. Probes

From design perspective these are intended to be light-weight components as these may typically reside in small devices such as end-user premise routers or customer premise equipment (CPE). In particular, these are devoid of any intelligent logic or interpretation/processing of data/result. From extensibility perspective, these are implemented as an array or list of plugins (functions), whereby each plugin is responsible for collecting and reporting data for a specific test scenario. Every probe knows its Configuration Server. Periodically, it contacts the Configuration Server for tests to run if any. Upon receiving test details, the probe component checks whether any plugins are missing and, if needed, fetches those from the Plugin Servers. The period of execution of a probe is dynamically configurable under the control of Configuration Server. The probe sends the collected raw data to Probe Cache Servers. A separate background process checks for any major software updates. Updates can also be governed by the Configuration Management Servers. Individual tests may be run in separate threads or independent processes depending on the hardware/software capabilities of the device and its impact on end user.

4.2. Configuration Management Servers

These are the brains of the measurement system. Service operators may distribute these geographically. These servers control which tests to run, which probes to run, how often to run etc. Through a randomization process, the impact of simultaneous access to these servers by probes may be minimized. Probes themselves do not have the capability to determine which tests are to be run periodically and which are to be run aperiodically. Configuration Management Servers can achieve the effect of periodic run by including the same test in each run request to probes. From design perspective, storage requirements would be low whereas access requirements would range from low to medium.

4.3. Plugin Servers

Operators may upload new plugins for new tests to Plugin Servers. The frequency of new test scenarios is assumed small. Thus, a single centralized server may do the job. For large operators, depending on customer base, these servers may be geographically distributed. From design perspective, storage requirements would be low whereas access requirements would range from low to medium.

4.4. Update Servers

These are contacted by probes periodically to check for any major software update. The assumption of the measurement tasks is that a major software update is a less frequent activity. Most updates would likely be in terms of new plugins that are handled by the Plugin Servers. Thus, in terms of design, high-availability is not a key requirement for these servers. If any update fails, then probes should at least be able to run with the current version. From design perspective, storage requirements would be low whereas access requirements would range from low to medium.

4.5. Probe Cache Servers

These are required to be able to handle both high-velocity as well as high-volume data. However, data integrity is not key since raw data is considered transient in nature and therefore may not need replication sets. From design perspective, both update/access and storage requirements would be high. Storage requirements may be made less stringent if an upper level entity purges the data once it has been fetched for processing and storage.

4.6. Refined Data Servers

These servers fetch raw data from Probe Cache Servers and process it. Processing may involve computing higher level metrics, trends, correlations, etc. These servers may keep historical data and may optionally purge raw data. Since upper level entities, such as a visualization layer, across the network would fetch data from these servers, a key requirement, if geographically distributed, is local consistency with guaranteed eventual global consistency.

5. Relationship to LMAP

The components of our software architecture given in Figure 1 can be mapped to the components of an LMAP-based measurement system. Functionally, the Probes component in Figure 1 is the same as Measurement Agents (MA) in LMAP terminology. The Configuration Management module is analogous to Controller in LMAP parlance.

The component labeled Probe Cache Servers is equivalent to LMAP's Collector. The directed edges between Probes and Configuration Management modules define the Control Channel described in the LMAP draft. There are some additional components in our architecture, for instance, Update Servers, Plugin Servers and Refined Data Servers, that are not part of the LMAP proposal. However, one can consider Update Servers as well as Plugin Servers as being part of the Controller's implementation details. Finally the Refined Data Server can be viewed either as a part of the Collector (if we assume that the Collector entity has data processing and mining capabilities), or as a separate upper-level entity that is not addressed by the LMAP framework.

6. IANA Considerations

There are no IANA considerations in this memo.

7. Security Considerations

This memo does not address security considerations related to a measurement system. It is assumed a fully deployed system will comply with the security considerations outlined in the LMAP framework and its subsequent revisions [RFC7594].

8. Acknowledgements

We wish to acknowledge the helpful contributions towards our implementation by Sandeep Vodapally, Carmen Nigro, Andrew Dammann, Eduard Bachmakov, Edward Gallagher, and Peter Rokowski.

9. Informative References

[Kulkarni2016]

Kulkarni, S., Gehlot, V., Brzozowski, J., Bachmakov, E., Gallagher, E., Dammann, A., and P. Rokowski, "A Scalable Architecture for Performance Measurement in Broadband Networks", 2015 IEEE Conference on Standards for Communications and Networking (CSCN), 2015, (to be published October 2015)", October 2015.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<http://www.rfc-editor.org/info/rfc1034>>.

[RFC7594] Eardley, P., Morton, A., Bagnulo, M., Burbridge, T., Aitken, P., and A. Akhter, "A Framework for Large-Scale Measurement of Broadband Performance (LMAP)", RFC 7594, DOI 10.17487/RFC7594, September 2015, <<http://www.rfc-editor.org/info/rfc7594>>.

Authors' Addresses

John Jason Brzozowski
Comcast Cable
1701 John F. Kennedy Blvd.
Philadelphia, PA
USA

Email: john_brzozowski@cable.comcast.com

Vijay Gehlot
Villanova University
Center of Excellence in Enterprise technology (CEET), Department of Computing Sciences, 800 Lancaster Avenue
Villanova, PA
USA

Email: vijay.gehlot@villanova.edu

Sarvesh Kulkarni
Villanova University
Department of Electrical and Computer Engineering, 800 Lancaster Avenue
Villanova, PA
USA

Email: sarvesh.kulkarni@villanova.edu