An SNMP Agent for Active In-network Measurements

G. Gardikis, K. Sarsembagieva, G. Xilouris, A. Kourtis
Institute of Informatics and Telecommunications
National Centre for Scientific Research “Demokritos”
Athens 153 10, Greece
{gardikis; katias; xilouris; kourtis}@iit.demokritos.gr

Abstract— **Active network measurements play an important role not only in network monitoring for OAM purposes, but also in assessing the current network status and providing real-time information for the optimization of applications. This paper proposes the extension of SNMP-based monitoring mechanisms to also support active in-network measurements conforming to the standardized OWAMP protocol. A specialized SNMP agent is designed and implemented, featuring an OWAMP-compliant probe module and accompanied by the appropriate MIB. The proper operation of the mechanism is validated in a laboratory testbed under emulated network conditions, achieving remarkable accuracy and seamless integration with SNMP-compatible network managers.**

Keywords- active measurements, OWAMP, SNMP, SNMP agent

I. Introduction

Given the diverse requirements of emerging user applications and the increased heterogeneity of contemporary networking technologies, efficient and accurate monitoring of the network infrastructure is becoming a more and more challenging task. Monitoring procedures have traditionally been used for OAM (Operations, Administration and Maintenance) purposes i.e. assessing the network health/status and detecting faults and suboptimal configurations. However, emerging application/network coupling paradigms, envisaged in the frame of the Future Internet introduce an additional perspective; they involve the (controlled) provision of the explicit knowledge of the network status not only to the infrastructure operator but also to the user applications, so that the latter can self-optimise, adapting to fluctuating network conditions. Such a concept is promoted e.g. by the Application-Layer Traffic Optimisation (ALTO) approach [1], [2].

Traditionally, network monitoring is mainly achieved via passive measurements i.e. observation of parameters such as link/interface status and load, queue lengths, routing tables etc. However, as will be explained later, active measurements, involving the insertion of probe traffic into the network, are also necessary for the accurate, end-to-end assessment of network conditions. In this context, we propose the extension of standardized SNMP-based network monitoring mechanisms so as to also support the conduction of in-network active measurements, in a way totally compatible with existing SNMP network managers. The aim is to have a common, integrated and standardized architecture for retrieving both active and passive metrics within the network i.e. without having to employ additional “measurement servers” at its edges.

The paper proceeds as follows: Section II provides some background information, necessary to better clarify the issues addressed. Section III describes the architecture and functionality of the enhanced “SNMP for Active Measurements” (SAM) agent and Section IV presents its proof-of-concept validation and assessment in an experimental testbed. Finally, Section V concludes the paper.

II. Background

As opposed to passive monitoring, which is based on the observation and recording of parameters and metrics retrieved from the network nodes during normal operation, active measurements involve the insertion of probe (i.e. artificially generated) traffic into the network. This artificial traffic has certain characteristics, such as packet size and rate, number of packets, inter-packet times etc. and is generated by a probe module. It is captured and measured both at the source and the destination node, and the measurement results reflect the status of the network path, which the traffic traversed. Since the probe traffic imposes additional load in the network, and may interfere with user traffic, it is usually restricted in volume and duration. Under this restriction, active measurements can be extremely valuable even on a Tier 1 operational IP backbone [3], complementing traditional, element-level passive monitoring. Active measurements can be particularly useful for assessing one- or two-way metrics for end-to-end paths, which cannot be derived via passive monitoring. Commonly collected end-to-end metrics are packet delay, loss, jitter/delay variation and reordering.

In order to yield realistic results, probe traffic must have similar characteristics to actual application streams. However, due to its restricted duration and volume, significant deviation between measured and actual values may occur. In [4], it is shown that very small packet loss ratios cannot be accurately measured with a short burst of probe traffic. On the other hand, delay measurements via time-stamped traffic can be quite accurate. Naturally, as explained in [5], the assessment of One-Way Delay (OWD) requires strict synchronization between the sender and the receiver; this can be partially achieved via NTP queries to an accurate time server or, even better, with GPS timing provided by satellite receivers.

When it comes to the mechanism for conducting active measurements and retrieving the results, many proprietary architectures have already been in used for a long time, such as [6], [7], the Test Traffic Measurement service (TTM) of RIPE [8], or several commercial tools developed by network tester manufacturers. The One-Way Active Measurement Protocol (OWAMP), described in RFC 4656 [9] is an effort to standardize this procedure, and is the protocol adopted by the proposed architecture described in this paper. OWAMP specifies the establishment of measurement sessions between two peers (Client and Server), the dispatch of the probe time-stamped UDP traffic and the subsequent exchange of measured quantities, such as one-way delay, loss and jitter.

For conducting network management and monitoring procedures, the Simple Network Management Protocol (SNMP), even after more than two decades since its introduction, is still the dominant standard, supported by almost all contemporary network elements. SNMP organizes the measured metrics into a hierarchical data structure called the Management Information Base (MIB). The module residing in the managed element (SNMP Agent) responds to the queries of the managing entity (SNMP Manager) by providing the values of certain objects in the MIB, corresponding to measurable quantities.

SNMP is widely being used to retrieve intra-element passive metrics, such as interface status and utilization, queue lengths, packet loss. However, there is a missing link between the SNMP and OWAMP standards, making it impossible to retrieve OWAMP-based active measurements via SNMP. We address this issue by proposing an integration framework between the two standards, built around an enhanced SNMP-for-Active-Measurements (SAM) Agent and a tailored MIB. The SAM Agent, in addition to reporting passive metrics, can be also used for conducting active measurements with a remote peer agent. It can be installed either in a network node (router) or in measurement servers at the edge of the network. The advantage is obvious; both active and passive measurements are integrated under the SNMP “umbrella”, making it possible for a standard SNMP-compliant network manager to configure and collect both types of metrics.

The following section presents the architecture and implementation of the proposed agent.

III. Agent Architecture and Functionality

A. SAM-MIB

At first stage, in order to support the conduction of in-network active measurements by the network elements themselves via SNMP, a new tailored Management Information Base needs to be designed and implemented. This MIB (“SAM-MIB”) needs to contain both:

- Configuration parameters for each OWAMP session to be launched: peer IP address, number of packets to be sent, packet size etc. Note that multiple OWAMP sessions may be run in parallel, where each session corresponds to a certain peer node and a specific path within the network which is measured (e.g. a Label Switched Path in the case of MPLS-enabled core).
- Results from each OWAMP session: number of hops traversed, packet loss, jitter and one-way delay (OWD), count of duplicate packets etc. Since the session consists of two unidirectional probe streams sent in opposite directions (from the OWAMP Client to the Server and vice versa), two measurement sets are available correspondingly.

Table I shows the SAM-MIB objects, organized in a table structure to support multiple simultaneous OWAMP sessions. Once the Network Manager configures each session (via SNMP SET commands), a dedicated Monitoring Daemon (i.e. a SAM Agent component, see following section) initiates the session automatically and executes it periodically and continuously in the background. At any time, the Manager can retrieve the most current results via a SNMP GET query.

TABLE I. OBJECTS OF THE SAM-MIB

Object name	**Description**
samDaemonRunning	Shows whether the daemon should be active or not. (a value of 0 suspends the active measurement procedure)
samDaemonInterval	Time Interval between two consecutive measurements (msec)
samActMsmtTable	Active Measurements and Configurations Table
└samActMsmtEntry	Row of the samActMsmtTable
└samActMsmtConfVPathId	(Configuration) Unique index corresponding to the network path measured.
└samActMsmtConfPeerIP	(Configuration) The IP Address (or Host Name) of the peer node with which the OWAMP session will be established
└samActMsmtConfNoPkts	(Configuration) Number of probe packets to be sent during the OWAMP session
└samActMsmtConfPktSize	(Configuration) Size of the packets to be sent during the OWAMP session
└samActMsmtConfInterPkt Time	(Configuration) Time interval between two consecutive packets (msec)
└samActMsmtConfLoss Timeout	(Configuration) Time threshold which, if exceeded, a packet will be considered lost (msec)
└samActMsmtOutboundNo Hops	(Result) Number of Hops measured in the Outbound direction(from the local node to the peer node)
└samActMsmtOutbound DuplicateCount	(Result) Number of Duplicate Packets measured in the Outbound direction
└samActMsmtOutbound PktJitter	(Result) Packet Jitter value obtained in the Outbound direction (msec)
└samActMsmtOutbound PktsLost	(Result) Number of packets lost in the Outbound direction
└samActMsmtOutbound MinDelay	(Result) Minimum one-way delay in the Outbound direction (msec)
└samActMsmtOutbound MaxDelay	(Result) Maximum delay value in the Outbound direction (msec)
└samActMsmtOutbound AvrgDelay	(Result) Average delay value in the Outbound direction (msec)

Object name	Description
└samActMsmtOutboundMedianDelay	(Result) Median delay value in the Outbound direction (msec)
└samActMsmtOutboundStDevDelay	(Result) Standard Deviation delay value in the Outbound direction (msec)
└samActMsmtInboundNoHops	(Result) Same as OutboundNoHops, in the Inbound direction (i.e. from the peer node to the local node)
└samActMsmtInboundDuplicateCount	(Result) Same as OutboundDuplicateCount, in the Inbound direction
└samActMsmtInboundPktJitter	(Result) Same as OutboundPktJitter, in the Inbound direction
└samActMsmtInboundPktsLost	(Result) Same as OutboundPktsLost, in the Inbound direction
└samActMsmtInboundMinDelay	(Result) Same as OutboundMinDelay, in the Inbound direction
└samActMsmtInboundMaxDelay	(Result) Same as OutboundMaxDelay, in the Inbound direction
└samActMsmtInboundAvrgDelay	(Result) Same as OutboundAvrgDelay, in the Inbound direction
└samActMsmtInboundMedianDelay	(Result) Same as OutboundMedianDelay, in the Inbound direction
└samActMsmtInboundStDevDelay	(Result) Same as OutboundStDevDelay, in the Inbound direction
└samActMsmtLastUpdate	The timestamp (Date and Time) of the more recent measurement results available

B. *SAM Agent*

The SNMP-for-Active-Measurements (SAM) Agent is essentially an integration of a standard SNMP agent with an OWAMP active probe. It is developed in C on a Linux Debian operating system. It is a composite module, consisting of four entities:

a) a standard SNMP agent (exploiting the linux net-snmp library) to ensure interaction with the SAM-MIB. A sub-agent was added to handle the SAM-MIB

b) an OWAMP client, which conducts the OWAMP sessions to a remote peer. The open-source reference RFC implementation [10] was used, modified so as to allow more fine-grained processing of the results (calculation of average One-Way Delay and OWD standard deviation from the set of per-packet OWD values)

c) an OWAMP server [10], for accepting and serving OWAMP session requests from other peers,

d) a dedicated Monitoring Daemon (developed from scratch) which coordinates the active measurement procedure.

The SAM agent also utilises an NTP client for achieving time synchronisation to an external NTP server, essential for accurate one-way delay measurements.

The monitor daemon is a module especially developed for performing and storing active measurements by invoking the OWAMP client component. Written in C, the monitor daemon auto-starts at device boot and runs continuously in the background. Its functionality is depicted in Fig.1. After initialisation phase, it obtains the SAM-MIB::samDaemonInterval and SAM-MIB::samDaemonRunning configuration values and examines their values. If the samDaemonRunning value is different than "1", then it waits for samDaemonInterval milliseconds and repeats the previous steps. If the samDaemonRunning is "1", then it proceeds to its main functionality, as follows.

Prior to executing any measurements, the monitor daemon synchronises the system by executing a NTP Time update query to a pre-configured time server. Then it retrieves all ConfVPathID/ConfPeerIP pairs from the samActMsmstEntry table. For each such pair, the monitor daemon spawns a child thread; a multi-threaded approach was chosen so as to be able to conduct measurement sessions to different peer IPs in parallel. Each thread retrieves the rest configuration parameters from the samActMsmtTable, which will be used to configure and execute the OWAMP session (number of packets, packet size etc.). Next, each thread invokes an instance of the OWAMP client, which connects to the peer OWAMP server so as to conduct the measurement session in both directions; two concurrent probe bursts are initiated, one inbound and one outbound. After the session is completed, the calculated measurements (no.of hops, jitter, loss, one-way delay, duplicates) are stored in the corresponding columns and rows of the samActMsmtTable.

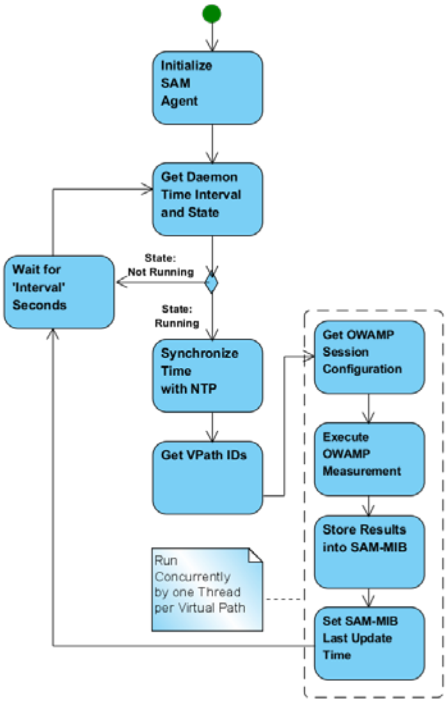
Initialize SAM Agent
Get Daemon Time Interval and State
Wait for 'Interval' Seconds
State: Not Running
State: Running
Synchronize Time with NTP
Get VPath IDs
Get OWAMP Session Configuration
Execute OWAMP Measurement
Store Results into SAM-MIB
Run Concurrently by one Thread per Virtual Path
Set SAM-MIB Last Update Time

Figure 1. Functional description of the SAM Monitor daemon

It must be noted, that, once configured, each OWAMP session is running repeatedly. In this way, the manager can always have instant access to the latest measurement data by executing an SNMP query to the SAM-MIB. The procedure is stopped when the SNMP manager removes the corresponding configuration parameters from the samActMsmtEntry table. Furthermore, the probe session configurations may be changed on-the-fly at any time and for any peer, without causing any disturbance to the other running threads/sessions.

In order to better clarify the overall operation, Fig.2 depicts a sequence diagram showing the interactions between all involved modules.

In order to facilitate wider experimentation and code improvement/enhancement, the SAM agent and MIB entities have been released as open-source code [11].

IV. VALIDATION

Following the design and the implementation of the SAM agent, we carried out a validation procedure in an experimental network testbed so as to verify the proper operation and accuracy of the SAM framework, along with its compatibility with standard SNMP network managers. The testbed we used is shown in Fig.3 and consists of:

- two linux-based edge routers (R1, R5), in both of which the SAM agent is installed.
- three linux-based routers (R2, R3, R4), forming the core network. These are deployed as Virtual Machines (VMs) within the same physical server. All links between R1-5 are 1GBps Ethernet connections (physical or virtual ones)
- a Network Manager, operating on a separate management network and featuring the open-source SNMP-based monitoring environment Zenoss Core [12], to which the SAM-MIB has been loaded as custom MIB; no further modifications are necessary. Via SNMP, the Network Manager supervises the network nodes (particularly R1 and R5) in a standardised way.
- a hardware-based NTP server, which provides accurate clock synchronisation to the system; it is reminded that tight sync is necessary for accurate one-way-delay measurements. The Stratum 1 NTP server has a satellite antenna and utilises GPS clock source. In addition, all Linux-based routers feature a precision-time kernel so as to support sub-msec clock accuracy. This configuration allows R1 and R5 to be in-sync with accuracy in the order of 0.1 msec.

Initially, the Network Manager configures the SAM agent in R1 to invoke an OWAMP session to R5. Apart from peer IP, the parameters configured via SNMP SET commands were:

- samActMsmtConfNoPkts -> 100 (number of packets to be used)
- samActMsmtConfPktSize -> 1000 (packet size, adding 14 bytes of minimum UDP payload)
- samActMsmtLossTimeout -> 2 seconds

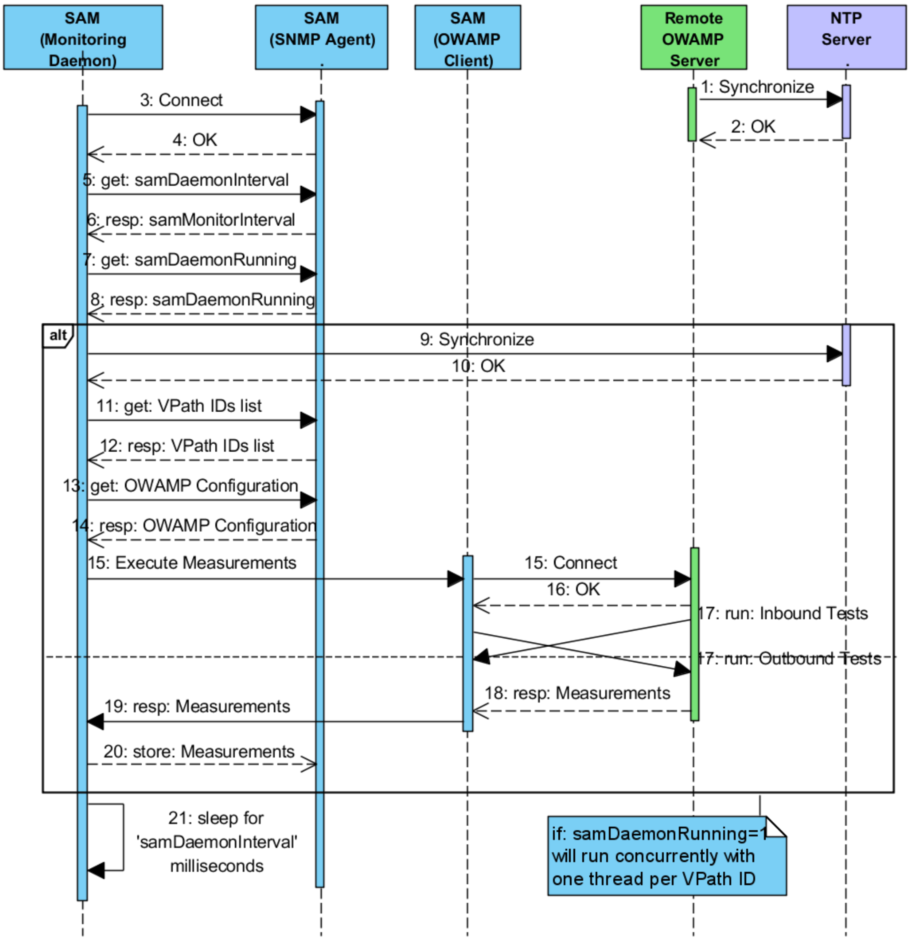

Figure 2. Sequence diagram of the active measurement procedure

- samActMsmtInterPktTime -> 0.1e (i.e. 0.1 seconds on average, exponentially distributed)

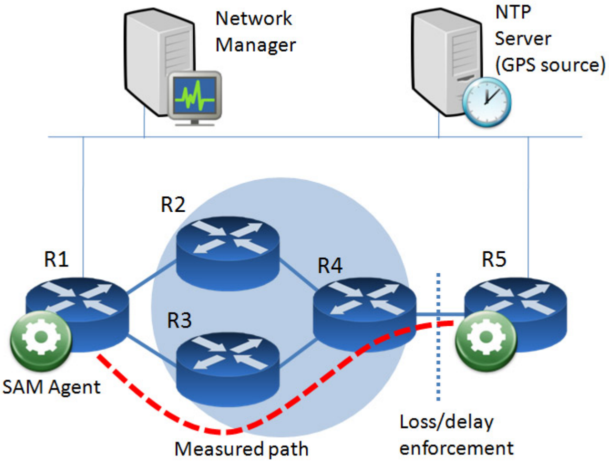

Figure 3. Testbed for assessing and validating the SAM framework

The network path used is {R1, R3, R4, R5}, as shown in Fig. 3. In order to emulate actual network conditions in a controlled manner, we enabled the linux built-in network emulator (netem) module in R4, on the R4/R5 network interface. Netem was used to apply a specific amount of loss and delay to the outbound (R4->R5) traffic.

After each change in the netem configuration, (OWAMP measurements were performed repeatedly) we executed SNMP GET queries from the Network Manager to R1 so as to observe how one-way delay and loss values were affected by the emulator. As a first experiment, we imposed delays from 0 to 50 msec with a 5-msec step. With the emulator disabled, the propagation delay added by the network itself is negligible - round-trip-time from R1 to R5 was in the order of 0.5 msec. Therefore, we just compared the measured values with the enforced ones; the relationship is depicted in the graph of Fig.4, where it is seen that the two values almost coincide, proving the accuracy of the measurement mechanism.

It must be noted that, since emulated delay was applied only to outbound traffic, Fig.4 refers to outbound delay only; in particular, the measured OWD corresponds to the values of the SAM-MIB::samActMsmtOutboundAvrgDelay object. The inbound OWD (not shown in the graph) is correctly measured to be approximately 0.2 msec - almost half of the “native” RTT of the network.

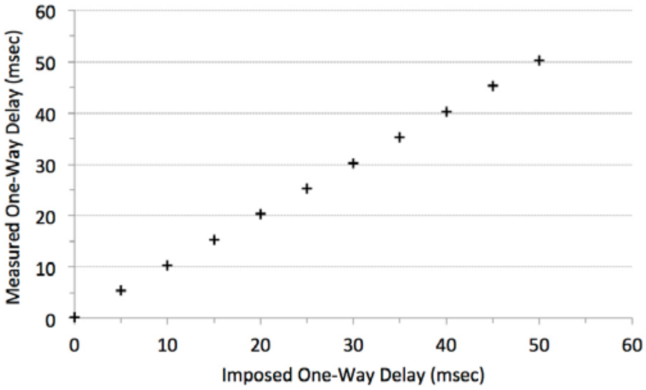

Figure 4. One-Way Delay: Imposed vs. Measured

At a second series of measurements, after setting the delay back to zero, we imposed packet loss on the outbound traffic. Loss values ranged from 0 to 5%, with a step of 0.5%. Uniform loss distribution was applied. Fig.5 shows that the values reported by the SAM agent achieve a very good approximation of the actual loss percentage, given the relatively limited length of the probe traffic burst (100 packets). Increasing the burst length would yield more accurate results but would also impose more overhead on the network.

Again, since emulated loss was applied only to outbound traffic, Fig.5 refers to outbound loss only; in particular, the indicated measured value corresponds to the ratio of the lost to the total outbound sent packets (SAM-MIB::samActMsmtOutboundPktsLost / SAM-MIB::samActMsmtConfNoPkts). The inbound loss (not shown in the graph) is correctly measured to be zero.

In a real operational network, the correlation between measured and actual loss will heavily depend not only on probe traffic parameters (packet numbers, measurement frequency etc.) but also on the mechanisms causing the losses (link errors, queuing disciplines etc.).

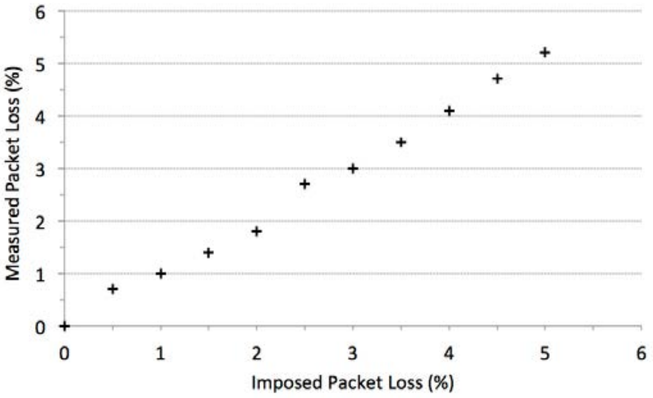

Figure 5. Packet loss: Imposed vs. Measured

V. Conclusions

The results from active measurements, taken along traffic paths on an operational network, are essential in order to form an accurate snapshot of the status and the condition of the network. In this context, the paper proposed a framework for integrating standardised procedures for executing active measurements (via OWAMP) and communicating configuration parameters and measurement results (over SNMP). The designed and implemented SAM agent, allows a common SNMP manager to jointly manage and exploit active measurements, along with passive ones. The proper operation of the agent was validated and its accuracy was assessed in a functional experimental testbed with emulated network conditions. To allow further experimentation and re-use, the developed agent components have been publicly released as open-source code.

Scheduled next steps include the integration of the modules in a composite network monitoring system combining active and passive measurements for the provision of "Network Distances" towards network-aware optimisation of applications, especially peer-to-peer ones. This integration is planned to take place in a more complex MPLS/DiffServ network testbed for measuring differentiated traffic on label-switched paths on end-to-end basis.

Acknowledgment

The work described in this paper is partially supported by the research project ALICANTE, funded by the European Commission within the Seventh Framework Programme under grant agreement n° 248652.

References

[1] J. Seedorf, E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, September 2009, http://tools.ietf.org/html/rfc5693

[2] R. Alimi, R. Penno, Y. Yang, "ALTO Protocol", Work in progress, March 2012, http://datatracker.ietf.org/doc/draft-ietf-alto-protocol

[3] L. Ciavattone, A. Morton and G. Ramachandran, "Standardized Active Measurements on a Tier 1 IP Backbone", IEEE Communications Magazine, June 2003, pp. 90-96

[4] P. Barford and J. Sommers, "Comparing Probe- and Router-Based Packet-Loss Measurement", IEEE Internet Computing, September 2004, pp. 50 -56

[5] L. de Vito, S. Rapuano, L. Tomaciello, "One-Way Delay Measurement: State of the Art", IEEE Trans. On Instrumentation and Measurement, 57(12), December 2008, pp. 2742-2750

[6] U. Hofmann, T. Pfeiffenberger, and B. Hechenleitner, "One-way-delay measurements with CM toolset," in Proc. Int. Performance Comput.Commun. Conf., Phoenix, AZ, 2000, pp. 41–47.

[7] J. Jeong, S. Lee, Y. Kim, and Y. Choi, "Design and Implementation of One-Way IP Performance Measurement Tool", vol. 2343. London, U.K.:Springer-Verlag, Jun. 2002, pp. 673–686. no. 2.

[8] RIPE Test Traffic Measurement Service, http://www.ripe.net/data-tools/stats/ttm/test-traffic-measurement-service

[9] S. Shalunov et al, "A One-way Active Measurement Protocol", RFC 4656, September 2006, http://tools.ietf.org/html/rfc4656

[10] Internet2, OWAMP version 3.3. http://www.internet2.edu/performance/owamp/

[11] SNMP for Active Measurements (SAM) Framework, http://www.medianetlab.gr/opensource/

[12] Zenoss Core, Enterprise IT Monitoring, http://sourceforge.net/projects/zenoss/