A Network Cost Provision Framework for Network-Aware Applications

Georgios Gardikis, George Xilouris, Katia Sarsembagieva, Anastasios Kourtis
Media Networks Laboratory
NCSR "Demokritos"
Aghia Paraskevi 15341, Athens, Greece
{gardikis; xilouris; katias; kourtis}@iit.demokritos.gr

Daniel Négru
CNRS- LaBRI lab
University of Bordeaux
Talence, France
daniel.negru@labri.fr

***Abstract*—While the explicit provision of Network Awareness to applications towards QoE optimisation has attracted significant attention during the past decade, very few solutions have been proposed which can be considered both feasible and scalable. Towards this direction, this paper proposes a framework which implements a Network Cost Service, explicitly providing the "network cost" between two or more network nodes as well as the a priori QoE estimation for a specific application - with the aim of assisting QoE-based optimization of current and emerging communication paradigms, such as peer-to-peer communication, distributed caching, information-centric networking and in-network cloud services. The proposed framework has been fully implemented, assessed and released as open-source for further experimentation. Large-scale deployment considerations and open research issues are also discussed.**

Keywords—Network-Aware Applications, Network Cost, QoE-based traffic optimisation

I. Introduction

The explicit provision of Network Awareness to applications is considered a promising aspect within the Future Internet scene, as a form of enabling resources awareness between the application and network domains [1]. Network Awareness involves the on-demand signaling of specific network-level metrics to applications, with the aim of optimizing application behavior and performance.

While such an interplay has been extensively studied by several research efforts, especially during the past decade, no universal solution has been proposed which can be scalable, widely applicable, and which respects the independence of the different architectural layers and also the privacy of the respective business actors.

Towards this direction, this position paper suggests a framework for providing explicit network awareness in the form of a network cost vector between two network nodes, accompanied by an application-specific a priori QoE estimation. The rationale and logic of this framework –based on an evolutionary architecture and leveraging existing trends in standardization bodies– aims at assisting traffic optimization at application layer, exploiting network status information, while at the same time addressing scalability, privacy and security issues.

The next sections of the paper better illustrate the addressed problem, describe the proposed framework, present evaluation results, and discuss a number of issues associated with "real-world" deployments.

II. Background and Motivation

For many decades, the success of the current Internet has been founded on the simple principle of host-to-host communication based on the functional logic of (mainly) isolated architectural layers. Due to this isolation, network-awareness is being only implicitly derived, mostly using in-band mechanisms, such as the TCP congestion control.

While some mechanisms have already been proposed for network-side explicit notification of network status and/or topology, often relying on centralized management entities for cross-layer interactions [2] [3] [4], most of them exhibit scalability issues which limit their large-scale applicability. Another aspect is the privacy of the network operator and the general unwillingness to publicly expose detailed network topology and utilization information. For these reasons, existing application-layer mechanisms, as described above, are still dominant for implicit network assessment.

However, as the Internet community is currently witnessing a gradual paradigm shift from host-centric to content-centric communication, existing implicit mechanisms may be not as efficient. In many current and emerging communication architectures (e.g. peer-to-peer, CDNs, in-network clouds, content-centric networks etc.), the content is distributed into the network, rather than concentrated in a single server. Therefore, its retrieval requires the simultaneous communication with tens, hundreds, or even thousands of peers. Assessing the communication with each of these peers at application level via the usual in-band methods would pose significant overhead in the network, besides requiring a considerable amount of time.

A more efficient alternative would be the establishment of an explicit Network Cost Service (NCS), offered by the network operator or a third-party entity and providing on-demand a "Network Cost" (NC) assessment for any given pair or group of network nodes. The Network Cost is a set of network-level parameters, which can directly mapped to an a priori QoE estimation for a specific application. Such an assessment could be instantly exploited at application level in order to optimize operations such as in-network content placement, peer selection, prioritization and load balancing.

A similar approach is promoted by the Application Layer Traffic Optimisation (ALTO) WG of the IETF, which, in addition to identifying the aforementioned need for a network cost service [5] is currently standardizing a protocol for communicating network cost maps, mainly focusing on optimizing peer-to-peer transfers [6]. ALTO assumes that these cost maps are provided by the network operators. The draft specification does not cover the procedure of deriving these cost maps but focuses on their formats and communication protocols.

Thus, the approach described in this paper considers the ALTO approach as a starting point and a) discusses and implements a complete framework including the provision of the network cost, leveraging active and passive measurement techniques, b) considers several other enhancements such as a more generic representation of the network cost, serving a much wider spectrum of use cases besides p2p and c) inherently supports the a priori (i.e. before application initiation) estimation of the expected Quality of Experience for various applications, based on the network cost metrics.

Candidate use cases for the proposed Network Cost Service include, but should not be restricted to:

- peer-to-peer file sharing and streaming - for optimizing p2p communication via selection of "best peers" i.e. peers with lower NC
- one-to-one and one-to-many real-time audiovisual communication – for a priori optimizing stream format and rate according to NC
- distributed caching, Content-Delivery Networks (CDN) and Information Centric Networking (ICN) – for choosing optimal locations for placing/caching content within the network
- in-network clouds – for efficient load balancing, traffic distribution and resource transfer to optimal locations, including Virtual Machine (VM) migration.

III. A NETWORK COST SERVICE FRAMEWORK

As aforementioned, Network Cost can be provided as a service either by the Network Operator (NO) or a third-party business actor. However, it seems that the more natural solution is that the NOs provide the service themselves, so that they can control the entire network measurement chain, from the network elements' monitoring agents up to the interface with the querying applications.

A high-level conceptual diagram is shown in Fig.1.

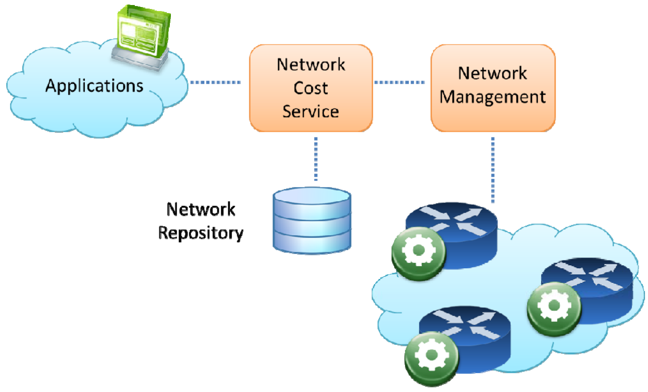

Fig. 1. High-level structure of a Network Cost Provision framework

The operation of a network cost framework would normally be restricted within a single administrative network domain (e.g. an ISP network or a backbone provider), although multi-domain scenarios could also be envisaged, as will be described. Within the network, existing management approaches can be used for collecting network metrics. To assist network cost estimation, it would be preferable that active measurements, in addition to passive ones, are employed. Active measurements include the controlled injection of artificial (probe) traffic across network paths so as to measure them. Probe traffic must be quite restricted in volume, so as not to induce considerable overhead in the operational network. The active measurement procedure could be undertaken by dedicated network probe modules, strategically placed within the network. Another option would be to enhance existing monitoring agents within the network elements so as to support the establishment of active measurement sessions on-demand. Our proof-of-concept implementation (described in next section) includes composite agents (tailored SNMP agents bundled with monitoring daemons) so as to perform active measurements on-demand, conforming to the OWAMP (One-Way Active Measurement) Protocol and supported by a custom-made MIB.

Another approach for retrieving active and passive network metrics would be via emerging SDN (Software Defined Networking) management mechanisms, where flow metrics would be directly retrieved from the SDN nodes by the SDN Controller using the Openflow protocol. In the same manner, the SDN Controller could also inject probe monitoring traffic into the managed nodes for conduction of active measurements.

The Network Cost Service itself receives raw network measurement data and assembles end-to-end costs, which are cached in a Network Repository for instant provision. The Repository should also host other network data, such as policies and topology, which assist in the cost estimation procedure. It must be noted that Network Costs are calculated by exploiting recently acquired measurement data already in the Repository; the invocation of a new in-network end-to-end measurement at each NC request would result in considerable overhead in the network and put the scalability of the proposed approach into question – in addition to raising security and stability issues.

The NCS exposes a public interface –commonly a SOAP-based or RESTful one- for providing costs on-demand, in e.g. XML or JSON format (For reference, the ALTO protocol uses JSON for more lightweight representation). Authentication and/or encryption mechanisms may also apply at this stage. Requests originating from applications should contain the addresses of the end-points between which the Network Cost is requested. For content/service-aware networks, which provide different treatment to various services (e.g. prioritise realtime streams against background traffic), Application Type information will also need to be explicitly signaled, since a given pair of endpoints will exhibit different costs for different applications.

For maximum flexibility, it is proposed that the network cost itself is provided as a vector, rather than a single scalar value:

$$\bar{C}^{(a,b)} = \left(c_1^{(a,b)}, c_2^{(a,b)}, c_3^{(a,b)}, \ldots, c_n^{(a,b)} \right)$$

where $c_i^{(a,b)}$, as a real number, denotes a certain network metric between endpoints a and b. For example, c_1 can correspond to hop count, c_2 to one-way delay, c_3 to average packet loss, c_4 to jitter/delay variation etc. These metrics can be derived via both active and passive measurements performed within the network. Additionally, information available from routing protocols can be exploited, such as EIGRP [7], which already uses vector structures to represent the "network distance" of a certain path [8].

Furthermore, in order to facilitate procedures such as peer ranking and selection, it is required to eventually collapse the cost vector to a single scalar value corresponding to the application-specific network cost, corresponding to a priori QoE estimation. The simplest approach for such a mapping involves a linear model, in which each application type corresponds to a weight vector W:

$$\bar{W} = (w_1, w_2, w_3, \ldots w_n)$$

Each element of the weight vector corresponds to the sensitivity of the specific application against a specific network metric, showing how the application experience degrades as the metric increases. Keeping the aforementioned correspondence, w_1 denotes the sensitivity of the application to hop count, w_2 to one-way delay etc. Thus, the application-specific network cost, as a scalar value, corresponding to the priori QoE estimation, can be expressed as the product of the two vectors:

$$NC^{(a,b)} = \bar{C}^{(a,b)} \cdot \bar{W} = c_1^{(a,b)} w_1 + c_2^{(a,b)} w_2 + \cdots + c_n^{(a,b)} w_n$$

The above expression assumes linear sensitivity to network conditions for simplicity. However, it is well known that, when it comes to estimating the perceived QoE and the Mean Opinion Score (MOS) from the network parameters, the linearity assumption is far from realistic; for example, in UDP video streaming applications, it is known that image quality degrades exponentially with packet loss [9]. In this case, more complex models could be employed, including nonlinear regression and neural networks [10]. The derivation of application-specific MOS from the network cost vector, to be demonstrated in the next section, indeed follows nonlinear models. However, since the purpose of the network cost is to be a quick and rough estimate of the communication quality between two endpoints and not a realistic approximation to the actual Quality of Experience, as perceived by the user, the aforementioned linear model could be adequate in most cases.

IV. Framework Implementation and Assessment

Within this section, we will present a reference implementation as well as a functional assessment of the propose framework.

Fig. 2 shows an integrated network testbed, incorporating a proof-of-concept implementation of the Network Cost Service framework, as it was designed and implemented in the frame of the EU FP7 ICT ALICANTE [11] project and is currently being extended in the frame of FP7 T-NOVA project. The technologies and protocols which were used for producing and communicating the cost metrics are only indicative, and can be replaced by equivalent ones.

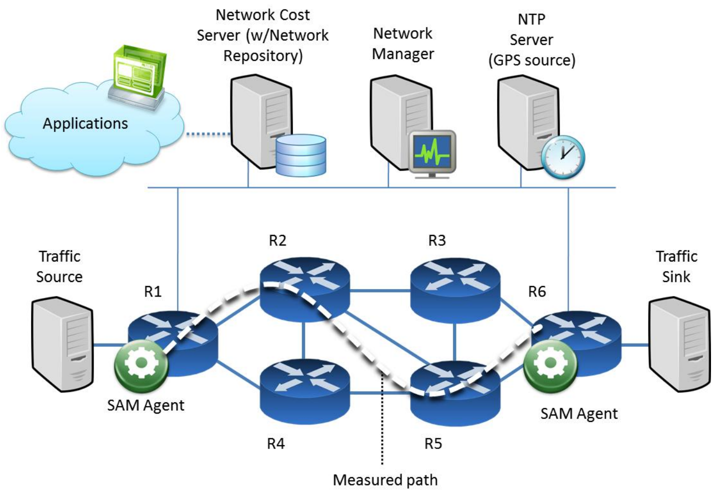

Fig. 2. Network testbed including a proof-of-concept implementation of the Network Cost Service

The experimental network domain of the testbed aims to emulate the core and edge network segments. It consists of six Linux-based routers (R1-R6), which constitute a DiffServ/MPLS domain. Each of the edge routers (R1, R6) hosts a specially developed composite agent (SNMP for Active Measurements – "SAM agent"). The SAM agent consists of a tailored SNMP agent plus a dedicated monitoring daemon, in order to support the establishment of active measurement sessions conformant to the OWAMP (One-Way Active Measurement) Protocol [12]. SAM agents utilize a custom-built MIB (SAM-MIB) and are configured via SNMP to periodically conduct active measurements. They communicate the measured metrics to the Network Manager also over SNMP. We have released the SAM Agent alone as open-source [13] and more technical details, including the SAM-MIB structure are to be found in [14].

The Network Manager configures the SAM agents to periodically conduct measurements to other elements in the network. The Network Cost Server retrieves the results and forms the end-to-end Network Costs to be communicated to applications via an exposed WebService. NCs are provided over the SOAP protocol in the form:

$$\bar{C}^{(a,b)} = \left(c_1^{(a,b)}, c_2^{(a,b)}, c_3^{(a,b)}, c_4^{(a,b)}, c_5^{(a,b)} \right)$$

where:

- $c_1^{(a,b)}$ is the number of hops traversed
- $c_2^{(a,b)}$ is the average one-way delay (OWD) experienced by the probe traffic packets (in μsec)
- $c_3^{(a,b)}$ is the average packet loss
- $c_4^{(a,b)}$ is the average jitter (in μsec)
- $c_5^{(a,b)}$ is the percentage of duplicate packets received

Accurate one-way delay (OWD) measurements [15] require tight synchronization between the sender and the receiver of the probe traffic. In our testbed, this is made possible thanks to a dedicated Stratum 1 NTP server, driven by a GPS clock and achieving μsec-order synchronization among the network elements. A preliminary assessment campaign, whose results are presented in [14], shows that the SAM agent is able to measure one-way delay and loss with remarkable accuracy, using only small probe traffic bursts.

Furthermore, in order to support proper monitoring of DiffServ-enabled networks, Network Cost queries also include traffic class information. In order to provide realistic measurements corresponding to the specific service, SAM agents mark the DSCP field accordingly in the probe traffic packets. Thus, the same pair of endpoints may yield different cost vectors for different applications.

In order to demonstrate this effect, in our tests, we used the NCS to retrieve the Network Cost Vector between R1 and R6 over the path shown in Fig.2 for several traffic classes. Within the testbed, we configured four DiffServ PHBs: EF (Expedited Forwarding, using pfifo queuing discipline), AF11/AF12 (Assured Forwarding, using GRED qdisc) and BE (Best Effort, using HTB/RED qdisc). An external traffic generator source/sink pair was used to heavily load the network with background traffic, emulating congestion conditions. The proper operation of the Network Cost Service under these conditions is depicted in Fig.3, which shows a visualization of Network Cost Vector samples between R1 and R6 for three different DiffServ classes: EF (set A), AF11 (set B) and BE (set C). Only three elements of the vector were used for visualization (i.e. one-way delay, loss and jitter), to enable projection in a three-dimensional space. As expected, under congestion conditions, the BE probe traffic exhibited high loss, the AF probe traffic experienced high delay, while EF traffic was forwarded with minimum OWD and loss.

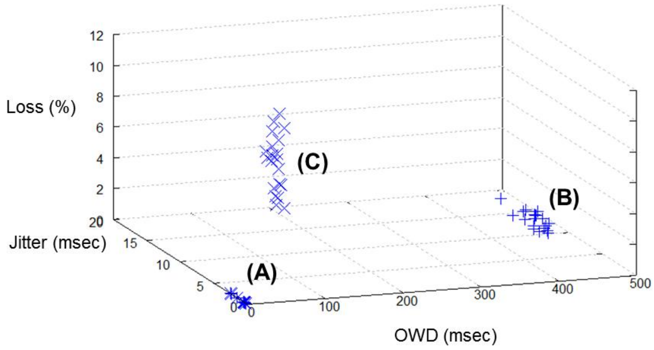

Fig. 3. Visualisation of Network Cost vector samples for three different classes: EF (A), AF11 (B), BE (C)

The next procedure to be carried out was the conversion of each network cost vector to a single scalar value, as described in the previous section. This conversion is application-dependent and corresponds to the derivation of the estimated QoE (MOS value) from the network parameters. We concentrated on three generic applications, namely Web browsing, VoIP and RTP/RTSP-based IPTV. For these three cases, the correlation between QoE and network state has been extensively studied in the literature. We chose three fairly simple approaches (both linear and non-linear) for proof-of-concept implementation, allowing realtime QoE estimation; VoIP MOS estimation is based on [16], IPTV MOS approximation is based on the findings of [17], while Web browsing MOS is derived based on methodologies described in [18], in conjunction with TCP throughput estimation. Common assumptions were used in the configuration of the MOS models, with regard to their parameters (e.g. audio/video codec, packet size, bit rate, Web page size, etc.). For applications other than the ones aforementioned (e.g. gaming), a corresponding tailored QoE model needs to be employed.

Fig.4. shows the cost vectors (A), (B) and (C) of Fig.3, averaged and translated into MOS values for the three application types. This MOS estimation feature enables the application to know the a priori estimate of the QoE for the communication between two network nodes.

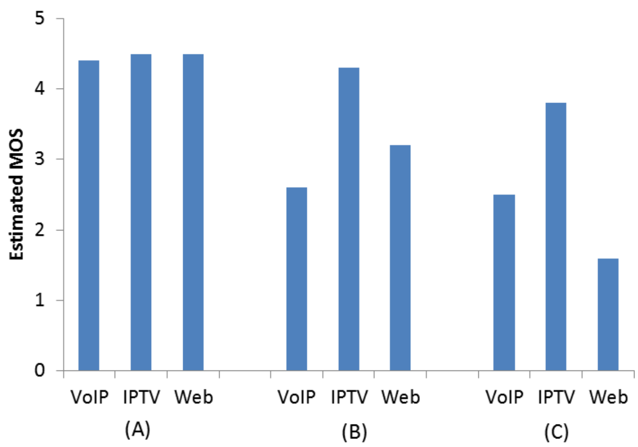

Fig. 4. Network Distance mapped to application-specific MOS

For the network-to-MOS mapping procedure, the respective nonlinear formulas from the aforementioned works were hard-coded into the Network Cost server. However, a more dynamic and reconfigurable approach, giving the opportunity to externally configure and fine-tune the MOS models, involves the formulas to be defined in an additional tables inside the Network Repository. A QoE estimator module can thus obtain the cost vector values and use the appropriate formula to calculate the MOS value (e.g. by using an equation parser, a formula evaluator or any other module/code). Going even further, a more complex approach, though the most dynamic, effective and scalable, could be the addition of a Table View or a Stored Function inside the Repository Database that would hold a complete, user defined, reconfigurable representation of any QoS-QoE mapping, in the form of a (complex or simple) SQL code. In this context, the Database could dynamically hold any type of QoE-QoS mapping, with virtually any type of formulae (linear, exponential, hybrid) that the system administrator could require or choose.

In order to facilitate further experimentation, we have released the integrated Network Cost Framework, consisting of the SAM agent, SAM MIB, Network Manager, Network Cost Server and Network Repository database (excluding QoE mapping), as open-source [19]. A detailed how-to guide is also included, describing the installation and operation of the framework in Linux-based network testbeds.

V. Other Considerations

When it comes to a real-world, large-scale implementation and deployment of a Network Cost Service, there are a number of considerations which must be taken into account:

A. Security and Network Operator Privacy

In addition to employing anti-DoS policies to avoid a denial-of-service attack to the NCS, it might be desirable that encryption and authentication mechanisms are employed in the NCS public interface, so as network costs are provided only to authorized clients. Furthermore, it is true that batch, automated NC requests would allow the recipient to gradually form a thorough snapshot of the network status, even exploiting techniques such as network tomography[20]. This could be a privacy breach for the Network Operator, since it would publicly expose sensitive network information. For this reason, certain restriction policies should apply even to authorized NC requests.

B. Inter-domain operation

The aforementioned architectural description, along with the proof-of-concept implementation corresponds to NC provision within a single administrative network domain. For multi-domain operation, the most realistic solution involves the cooperation between NC Servers of adjacent domains. Thus, a NC estimation to a peer belonging to a foreign domain should result from the communication of the NC Servers from all intermediate domains involved in the network path. This communication should probably take place in a cascading form, while the final end-to-end metrics should result as an assembly of the inter-domain ones (e.g. adding the OWDs etc.)

C. Access network issues

The proof-of-concept implementation which was described is mainly applicable to the core end edge network segments. It is however essential to also include the access network in the measurement procedure, since bottlenecks often occur in the access part. However, performing end-to-end measurements among thousands or millions of user terminals (mostly CPE/home gateways) would raise major scalability issues. For this purpose, it could be proposed that the link between the CPE and the access router is evaluated separately, via a WAN management protocol such as [21] and then combined with in-network path measurements so as to form the end-to-end cost. This of course complicates the functionality of the NC Service, but could yield a feasible solution.

All the aforementioned issues constitute interesting fields for further research.

VI. Conclusions

Within the Future Internet scene, which is shifting from host-centric to content-centric communication, a Network Cost Service -explicitly providing a set of network metrics characterizing the path between two network nodes- seems a promising approach, offering applications the capability to automatically self-optimise according to network conditions. We presented a generic yet feasible framework for providing a Network Cost Service and described a fully functional proof-of-concept implementation, released as open-source. Preliminary results show that Network Costs can be provided with relatively minimal overhead in the network operation, and exhibit remarkable accuracy. Next research tasks should include experimenting in wider-scale networks and also migrating the Network Cost framework to SDN-based management architectures, also dealing with security, privacy, inter-domain and access network issues, as mentioned in the previous section.

Acknowledgments

This work has been supported by the research project ALICANTE, funded by the European Commission within the Seventh Framework Programme under grant agreement n° 248652 and is being continued in the frame of EC FP7 T-NOVA project, funded by the EC under grant agreement n° 619520.

References

[1] European Commission, FIArch group, “Future Internet Design Principles”, January 2012, http://ec.europa.eu/information_society/activities/foi/docs/fiarchdesignprinciples-v1.pdf

[2] G. Gardikis, L. Boula, G. Xilouris, A. Kourtis, E. Pallis, M. Sidibe, D. Negru, “Cross-layer Monitoring in IPTV Networks”, IEEE Communications Magazine, Vol.50(7), June 2012, pp.76-84

[3] N. Miller, P. Steenkiste, "Collecting network status information for network-aware applications," in Proc. INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies., pp.641,650 vol.2, 2000

[4] J. Arnaud, D. Négru, M. Sidibé, J. Pauty, H. Koumaras, “Adaptive IPTV services based on a novel IP Multimedia Subsystem”, Multimedia Tools

& Applications, Springer, Vol. 55(2),333-352, DOI: 10.1007/s11042-010-0576-1, November 2011

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

[6] R. Alimi, R. Penno, Y. Yang, “ALTO Protocol”, Internet Draft (Work in Progress), http://tools.ietf.org/html/draft-ietf-alto-protocol-24, December 2013

[7] Cisco Systems, “Introduction to EIGRP”, Document ID: 13669, http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080093f07.shtml

[8] B. Albrightson, J.J. Garcia-Luna-Aceves, J. Boyle, “EIGRP – A Fast Routing Protocol Based on Distance Vectors”, in Proc. Networld/Interop 94, 1994

[9] H. Koumaras, C-H Lin, C-K Shieh, A. Kourtis, “A Framework for End-to-End Video Quality Prediction of MPEG Video”, Journal of Visual Communication and Image Representation, 21(2), February 2010, pp. 139-154

[10] C. Viard-Gaudin, D. Barba, “A Convolutional Neural Network Approach for Objective Video Quality Assessment”, IEEE Trans. on Neural Networks, 17(2), September 2006, pp. 1316-1327

[11] FP7/ICT ALICANTE: Media Ecosystem Deployment through Ubiquitous Content-Aware Network Environments, http://www.ict-alicante.eu/public/

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

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

[14] G. Gardikis, K. Sarsembagieva, G. Xilouris, A. Kourtis, “An SNMP Agent for Active In-Network Measurements”, in Proc. ICUMT 2012, October 3-5, 2012, St. Petersburg, Russia

[15] 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

[16] R. Cole and J. Rosenbluth, “Voice over IP Performance Monitoring”, ACM SIGCOMM Computer Communication Review, Vol. 31 (2), April 2001, pp.9-24

[17] G. Gardikis, G. Xilouris, E. Pallis and A. Kourtis, “Joint Assessment of Network- and Perceived-QoS in Video Delivery Networks”, Telecommunication Systems, Volume 49, Issue 1 (2012), p. 75-84

[18] J, Shaikh, “Non-Intrusive Network-Based Estimation of Web Quality of Experience Indicators”, Licentiate Dissertation, Blekinge Institute of Technology, 2012, ISBN: 978-91-7295-233-1

[19] Network Cost Service Framework, http://www.medianetlab.gr/opensource/

[20] R. Castro, M. Coates, G. Liang, R. Nowak and B. Yu, “Network tomography: Recent developments”, Statistical Science, Vol. 19, No. 3. (2004), pp. 499-517, doi:10.2307/4144399 Key: citeulike:4738893

[21] Broadband Forum, TR-069 CPE WAN Management Protocol, May 2004, http://www.broadband-forum.org/technical/download/TR-069.pdf