"Performance Evaluation of Mobile Agent Technology for the Configuration and Auditing of SLSs in DiffServ Enabled IP Networks"

Theodore Kotsilieris[1], Stelios Kalogeropoulos[1], George Karetsos[1], Vassilis Loumos[1] and Eleftherios Kayafas[1]

[1]*National Technical University of Athens, Division of Communications, Electronics and Information Engineering, Department of Electrical and Computer Engineering,*
9 Heroon Polytechneiou Str., GR-15773 Athens, Greece, tel:+30-2107721447,
{tkots, skalog@central.ntua.gr, karetsos, loumos, kayafas@cs.ntua.gr}

Abstract - Nowadays we witness a growing interest for the efficient management of the QoS delivered by the service providers to the Internet users and the usage of SLAs. In such an environment the need to maximise customer satisfaction and reliability and consequently, the deriving obligation to quantify and measure service functions becomes apparent. As traditional centralized NMS manager-agent approach has stretched to its limits, we propose a way of bringing together the static nature of SNMP protocol with the Mobile Agent Technology (MAT) in IP network. In order to address lack of quality of service inherent in current Internet technology, we have chosen differentiated services architecture by implementing the corresponding DiffServ MIB proposed by IETF. The major objective of this paper is to compare the required time to configure and monitor DiffServ domain routers by using MAs and the client-server model. The results drawn from our research lend support to the claim that the performance enhancement by adaptating MAT to rigid network management area depends mainly on the traffic load on the link connecting the managed network with the manager.

I. Introduction

The standard today's Internet Protocol (IP) [13] provides *best effort service* by default. Traffic is processed as quickly as possible, all packets receive the same quality of service and are forwarded using a strict queuing discipline but there is no guarantee referring to actual time of delivery. Despite the fact that transmission delays do not adversely affect typical Internet applications such as e-mail, file transfer and Web-based applications, they hold back applications with real-time requirements such as multimedia and VoIP. In support of emerging advanced networked applications, the current best effort IP must transition to a service model that will provide service alternatives to the users. Towards this direction, the IETF Differentiated Services Working Group proposed the DIFFerentiated SERVices (DIFFSERV) [1] architecture. In this scalable mechanism packets are classified and marked to receive a particular per-hop forwarding behavior on nodes along their path and assigned to different forwarding behavior aggregates. Each behavior aggregate is identified by a single DS code point (DSCP) which is placed on each packet's header [2].

Differentiated services are extended across a DiffServ domain boundary by establishing a Service Level Agreement (SLA) between an upstream network and a downstream DS domain. An SLA is a service contract committed between a customer and a service provider that specifies the forwarding service customer should receive. On the ground that SLA is a formal contract, in order to be determined, it has to be translated into pragmatic network parameters. That is where Service Level Specification (SLS) crops up. It specifies the technical aspects of a particular service that can be experienced by the customer, across a boundary domain link

In this work, we focus in presenting a prototype implementation for efficient SLS enforcement in IP networks by proposing a decentralized network management architecture based on the Mobile Agent Technology (MAT). This approach is compared with the classic client-server paradigm, considering as a metric the required time to configure and monitor a set of DiffServ domain routers through SNMP. For this reason an implementation of the DiffServ MIB for a Linux router is exploited [10]. The benefit will be to enable Differentiated Services management with SNMP features. Furthermore, our study is also focused on the issue of Mobile Agent scalability in Network Management applications. Based on a thorough analysis of the results we export some crucial conclusions for the improvement of the management's efficiency.

Unfortunately, the centralised manager-agent model (implemented by SNMP and CMIP) has failed to respond efficiently to the evolution and proliferation of communication networks and their increased requirements for scalable Network Management operations [5].

Mobile agents is an emerging technology that is gaining momentum in the field of the decentralized computing and affords new opportunities for the distribution of processing and control in NM area, promising to give solutions to the limitations of client-server approach. The benefits gained from the use of MAs in the Network management area [6] can be condensed in the following:

- Reduction in network traffic: The transfer of mobile agents to the data sources creates less traffic than transferring the data and the overall interaction has local character.
- Efficiency and space savings: Mobile agents reside and execute at one node at time.
- Robustness and fault tolerance: MAs can interact asynchronously without loss of accuracy
- Increased responsiveness: Because MAs reside near network elements, they can respond to network events, avoiding delays caused by network congestion. When failure occurs, agents communicate in order to reconfigure the network.

The remainder of this paper is organized as follows. Section II presents the detailed information comprising an SLS in a template form along with an overview of the DiffServ schema main components. Furthermore, we illustrate router's configuration in DiffServ domain and present our architectural model. Section III describes the Qos configuration and auditing application based on mobile agents and is compared on architecture basis with the classical client/server approach. Section IV comprises the network environment that was set-up and is followed by section V that presents the results attained, accompanied by a performance evaluation report. Section VI presents the conclusions and possible future directions.

II. Service Level Specifications and the Differentiated Services architecture

As it has been mentioned before, all levels of service provided by the DiffServ architecture [1] are defined by sets (such as Bandwidth, Flow Identification, Service Schedule, Excess Traffic Treatment, Performance Parameters, Traffic Profiles) of objective, measurable and meaningful parameters (such as Mean/Max/MinBandwidth, ServiceOrigin/Destination etc.) having thresholds set upon them and all these attributes are described in SLS (Service Level Specification) templates.
The block diagram of a classifier and traffic conditioner in Differentiated Services architecture is presented in figure 1.

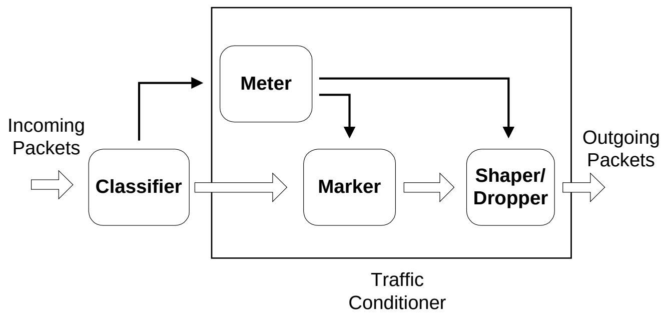

Figure 1: Packet Classifier and Traffic Conditioner in DiffServ

The components of this architecture are described below.
Packet Classifiers: They provide identification of the packets belonging to a traffic stream that may receive differentiated services.
Traffic Conditioners: They apply conditioning functions on the previously classified packets according to a predefined Service Level Specification (SLS).
Meter: Measures the temporal properties of a traffic stream selected by a classifier.
Marker: Sets the DS Codepoint in a packet based on well-defined rules.
Shaper: Delays packets within a traffic stream to cause the stream to conform to defined traffic profile.
Dropper/ Policer: Discards packets based on specified rules.

III. The QoS Configuration and Auditing Application of the system

Our SLS configuration and monitoring management system architecture highlights the very simple nature of the SNMP protocol combined with the emerging Mobile Agent Technology (MAT). All the SNMP functions (i.e get, getNext, and set) are performed by mobile agents (MAs) through Java SNMP API [16].

The MAs are able to migrate to the nodes where DiffServ SNMP Agents reside, perform their tasks and report to the Network Manager. Mobile agents decentralize processing and consequently reduce the traffic around the management station link. Also turn asynchronous the agent-management communication (useful when there exist unreliable or lossy links) and distribute the processing load increasing the flexibility of the management agents' behavior.
The above-mentioned advantages were evaluated on a testbed emulating the Mobile Network Manager case [12, 11] over a WAN. The SLS configuration and management system is composed of two modules: the GUI and the translator. The detailed operation of this system will be explained in the following section.
Each QoS parameter value is the outcome of computations upon MIB entries. In the SLS configuration/monitoring case, the Network Manager is only aware of it's high level specifications (e.g. delay, jitter, throughput, and bandwidth). A Graphical User Interface allows the Network Manager to specify the QoS parameters, necessary thresholds, and the type of operation to be performed by the QoS application.

The most representative operations that can be performed from our DiffServ configuration and monitoring application are described below.

- Create_Connection(in SLS_Template, out SLS_id)

This function allows a user to create a connection with a certain QoS between an origin and destination end point. This operation requires configuration of the classification and metering modules of the edge router of the connection. Also, in the case that the end point of the connection belongs to a different domain, configuration of border routers between the two domains is also needed.

- Destroy_Connection(in SLS_id)

This function allows a user to destroy a connection with a certain QoS between an origin and a destination end point. This operation requires configuration of the classification and metering modules of the edge router of the connection. Also, in the case that the end point of the connection belongs to a different domain configuration of border routers between the two domains is also needed.

- Set_Class_Values(in Class_Type, in Dscp_Value, in Bandwidth)

This function associates a specific class with a DSCP and minimum bandwidth values in the core routers of the network.

- Monitor_ClassesOfService (in Class_Type, out Packets_Sent, out Packets_Dropped)

This function gets for all routers of the domain the number of packets sent and the number of packets dropped for a specified class.

- Monitor Connection (in VprPid, out status)

This function gets for the specified VPrP the current status. This can be ACTIVE, PENDING, or DISALLOWED. Furthermore if the status is ACTIVE the network manager can also monitor several other aspects concerning the conformance of the connection to the contracted SLA with the user.

These operations require configuration and monitoring of SNMP variables of the edge and core routers. A software module able to translate from/to MIB entries was necessary to be included in our system. This module receives as input the high level parameters from the network manager and converts them into MIB variables. The translator module after performing conversion of the operations into a list of SNMP actions proceeds according to the selected operation mode (i.e. mobile agent or pure SNMP). The case of including such a functional layer as a wrapper on the managed nodes was rejected for efficiency reasons. By this way translation is performed only once.

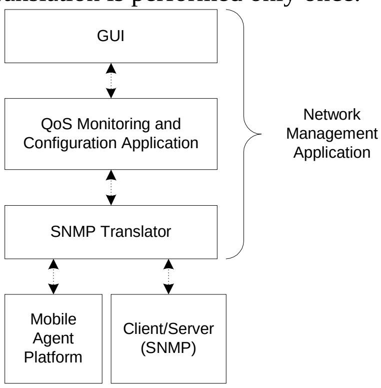

Figure 2: The Network Management application High Level Architecture

The modes of operation, namely MAs or Client-Server, are explained in detail in Section V. In brief the following juxtaposed experiments outline the performance comparison of mobile agents versus pure SNMP.

IV. Network Environment

This section is devoted to the network topology we employed to assess the performance gains of our system architecture with the fundamental SNMP.

In our experimental setup we performed configuration and monitoring management over WIN2000 Workstations implementing the Differentiated Services MIB [9]. In order to verify the optimisation performed by our system, we set-up a 100Mbps Ethernet LAN which played the role of the managed network, composed of three WIN2000 Workstations running on Pentium-III with 128MB of RAM. Each workstation was enabled with the Windows SNMP service and the Voyager MA platform [14]. Apart from the managed nodes, another PC was attached on the LAN acting as the Mobile Network Manager. The following figure depicts the testbed we have used.

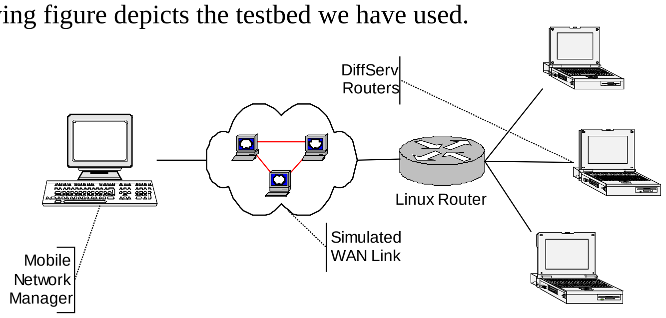

Figure 3: The network environment emulating the Mobile Manager case

The objective of our measurements' was the collection of the whole DiffServ-MIB tree statistics consisting of 960 distinct variables. Although the MIB variables are only 65, their number is multiplied by the number of network interfaces and the number of components as: classes, schedulers and filters. The operations performed for monitoring purposes are get and getNext. We performed 100 experiments first by using the Mobile Agent Technology (MAT) and then based on SNMP. The loaded link simulation was obtained using the Iperf [15] traffic generator. The generated traffic on the 100Mbps link between the network management station and the Linux Router varied from 0 Mbps to 100Mbps.

V. Experimental Results

The results acquired through the tests were processed thoroughly in order to assess the mobile agents' performance. According to the scheme adopted in our implementation a single mobile agent is created that traverses all the managed nodes. The SNMP (client/server) mode of operation followed the same scheme. The management operations occurred in serial mode. That is every managed node was queried one after another.

A mobile agent is initiated containing the Object Identifier of the MIB entries that should be get/set in order to monitor/configure the appropriate nodes. The travelling path is also determined and included in the agent. Upon the arrival of the MA on a workstation, the tasks addressed to each node are carried out through the DiffServ SNMP Agent. After concluding its trip, the MA returns back to the translator reporting the current status and the results from its actions. The architecture of our system is depicted in Figure 4. This architecture was compared with the classical client – server model that SNMP is based on. In the Client Server case, the Network Operator performs the tasks as explained above (i.e. proceeds with the configuration/monitoring of the next node after concluding the remote interactions with the current node). The Network Management station communicates directly with the DiffServ SNMP Agent. The communication was enabled through the usage of the AdventNet SNMP ver3.0 library [16].

As a first indication of their behavior, **Figure 5** illustrates the elapsed time for processing accepted requests as a function of cross traffic rate for the two technologies examined.

Response time results show that the mobile agent is less sensitive to the latency and the bandwidth of a bottlenecked link that connects the management station with the network nodes. This can be easily attributed from the fact that MAs do not need to cross the congested link more than twice (once it is injected and finally when it returns back to the manager). Keeping in mind that the traffic over the overloaded link must be kept as less as possible the Mobile Agent Technology appears to perform better than pure SNMP.

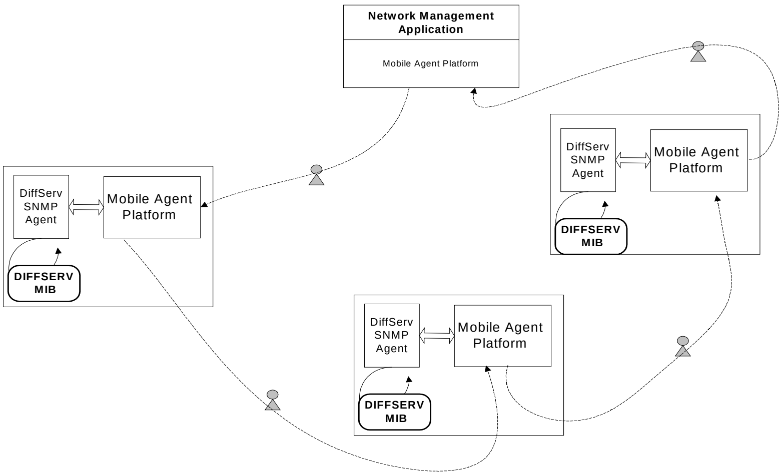

Figure 4: Monitoring / Configuration Architecture using MAs

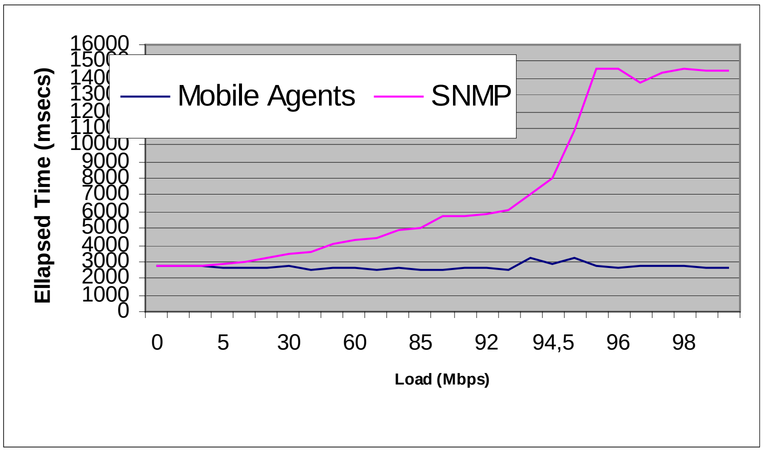

Figure 5: Comparison of the Client Server model vs. the MAT

The main conclusion exported from the above results outlines the performance improvement using the MAT. As the traffic on the link increases, the performance improvement using the MAT increases too. The only exception is in the case of negligible traffic (up to 2Mbps) where the two methods perform in quite the same way. As the cross traffic increases (>10 Mbps) the response time for the SNMP grows faster than the response time of the mobile agent, since the messages exchanged increase proportionally. For the mobile agents' approach the response time increases slower as the number of transactions grows due the steady size of the mobile agent. Another point that is worth to be mentioned is that the performance improvement for over 95.5 Mbps of cross traffic is almost stable (for 95.5 Mbps the improvement is 81.7% while for 100 Mbps is 80.84%).
A careful study on the generated traffic reveals the following: each get-request is exactly 86 bytes per host. Additionally, 50 bytes of protocol headers are added. The response to this is about 90+50 bytes. This results to 276 bytes transmitted for every request. The volume of data generated from the network management application is:

$$
\begin{aligned}
Traffic_{average} &= Variables * Bytes / request \Rightarrow \\
Traffic_{average} &= 950 * 276 \cong 264 KBytes
\end{aligned}
\qquad (1)
$$

The total traffic is the aggregate of the traffic generated by each node.

$$TotalTraffic = \sum_{i=1}^{n} Tr_i \text{ , n is the number of managed nodes} \qquad (2)$$

Due to the fact the number of monitored variables is the same for all the nodes we can claim with no loss of generality that the generated traffic equals to the traffic implied by a single node multiplied by the number of nodes.

$$TotalTraffic = n * Traffic_{average} \qquad (3)$$

VI. Conclusions and future work

We have presented the architecture and implementation of a decentralised network management system based on the Mobile Agent Technology. The research reported in this paper focuses on the time required for a mobile agent to accomplish a specific management task on a sequence of visited network nodes. The amount of management tasks requested and the cross traffic over the links connecting the mobile manager with the managed nodes are critical for the performance impact. As presented above, the performance improvement increases rapidly until a saturation point at around 95.5 Mbps.

The outcomes of this work are encouraging enough to allow us focus on other areas that could induce further performance gains. This is quite a promising outcome, since we haven't considered possible delegation of the single agent's tasks in more agents (even one MA per managed node) dedicated to specified operations, a fact, which most probably would lead in further performance improvement.

References

[1] S. Blake et. al., "An Architecture for Differentiated Services", RFC2475, December 1998.

[2] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the differentiated services field in the IPv4, IPv6 headers", RFC 2474, December 1998.

[3] W. Almesberger, J. Salim, A. Kuznetsov, "Differentiated services on Linux", Globecom '99, Rio de Janeiro, December 1999, pp. 831–836.

[4] J. Case, M. Fedor, M. Schoffstall, J. Davin, "A Simple Network Management Protocol (SNMP)", RFC 1157, May 1990.

[5] K. Meyer, M. Erlinger, J. Betser, C. Sunshine, G. Goldzsmidt, and Y. Yemini "Decentralising Control and Intelligence in Network Management", In Proceedings of 4th International Symposium on Integrated Network Management, Santa Barbara, CA, May 1995.

[6] Andrzej Bieszczad, "Mobile Agents For Network Management", IEEE Communications Surveys, Fourth Quarter 1998.

[7] Bandwidth Broker Implementation Version 1.0, Specifications for the Internet2-Qbone –bb Operability event, University of Kansas, Lawrence Septembesr 1999.

[8] K. Nichols, V. Jacobson, L. Zhang, "A two-bit Differentiated Services Architecture for the Internet" RFC2638, July 1999.

[9] F. Baker , K. Chan, A. Smith "Management Information Base for the Differentiated Services Architecture" INTERNET-DRAFT, February 2001

[10] Theodore Kotsilieris, Panagiotis Zikos, Efstathios Vlachos, Stelios Kalogeropoulos, Angelos Michalas, George Karetsos, Vassilis Loumos "A Prototype SNMP Management Framework for DiffServ Linux Routers, its Implementation and Performance" ,Net-Con'2002, Paris

[11] Sahai A., Morin C., Billiart. S *" Intelligent agents for a Mobile Network Manager (MNM) "* presented and published in the proceedings of the IFIP/IEEE International Conference on Intelligent Networks and Intelligence in Networks (2IN'97), September 1997, Paris, France

[12] Sahai A., Morin C. *" Enabling a Mobile Network Manager through Mobile Agents;* In Proceedings of the Mobile Agents'98

[13] http://www.qosforum.com/, QoS Forum White Paper, "QoS protocols & architectures", August 1999

[14] www.objectspace.com

[15] http://dast.nlanr.net/Projects/Iperf/

[16] www.AdventNet.com