

A Novel QoS Framework Based on Admission Control and Self-Adaptive Bandwidth Reconfiguration

A. Peculea, B. Iancu, V. Dadarlat, I. Ignat

Adrian Peculea, Bogdan Iancu, Vasile Dadarlat, Iosif Ignat

Technical University of Cluj-Napoca

Romania, 400020 Cluj-Napoca, 15 Constantin Daicoviciu

E-mail: {Adrian.Peculea,Bogdan.Iancu,Vasile.Dadarlat,Iosif.Ignat}@cs.utcluj.ro

Abstract: This paper proposes a novel end-to-end QoS framework, called Self-Adaptive bandwidth Reconfiguration QoS framework (SAR). SAR provides end-to-end QoS guarantees on a per-flow basis through admission control and end-to-end bandwidth reservation. In order to adapt to short and long time traffic load changing, SAR performs dynamic bandwidth reconfiguration. Due to a new organization of the network physical lines, SAR allows for a better utilization of the links' capacity and a smaller number of rejected flows, increasing the network's availability.

Keywords: end-to-end QoS, admission control, bandwidth reconfiguration.

1 Introduction

Computer networks transport simultaneously several flows, fact that makes necessary a multiplexing mechanism. Transport procedures affect the traffic flows, reason for which the traffic has to be characterized and quality of service (QoS) requirements need to be established. Traffic types and their QoS requirements impose the implementation of QoS methods and architectures. This paper presents the design and implementation of a new end-to-end QoS framework with self-adaptive bandwidth reconfiguration.

Integrated Services (IntServ) [1] provide end-to-end quality of service (QoS) guarantees for individual flows by maintaining the state and by reserving bandwidth for each flow at routers on the path between source and destination. The additional loading introduced by the per-flow bandwidth reservation processing and by the per-flow state maintaining at each router is significant and is increasing along with the network. For this reason, Integrated Services presents scalability problems.

Differentiated Services (DiffServ) [1] group the flows in traffic classes at the edge of the network. Interior routers forward each packet function of the per-hop behavior associated to the traffic class of the packet. Because of the flow aggregation and the lack of admission control, Differentiated Services do not provide end-to-end QoS guarantees to individual flows.

On-Demand QoS Path (ODP) [2] provides end-to-end QoS guarantees to individual flows introducing an additional load much lower than in the case of Integrated Services and maintaining a similar scalability to the one of the Differentiated Services. ODP exercises per-flow admission control and end-to-end bandwidth reservation at the edge of the network. Inside the network ODP differentiates the traffic classes as in the Differentiated Services. The main disadvantage of ODP is that the bandwidth adjustment is only inside the traffic class and does not allow for bandwidth redistribution between classes. The free bandwidth of the Provisioned Links that are not used or present a low utilization can not be made available for other Provisioned Links, the free bandwidth remaining unused. Another disadvantage of this framework is the fact that it does not include a module for determination of the bandwidth necessary for each input flow.

In order to eliminate the disadvantages above mentioned, we elaborated, implemented and proposed a framework for end-to-end quality of service guaranteeing through admission control

and self-adaptive bandwidth reconfiguration, which allows for bandwidth redistribution between classes. In this approach, the Physical Line is divided into two main sections, a part being the Guaranteed Link (GL) necessary for guaranteeing a minimum bandwidth (where is the case) for traffic classes (TCs), and a common part named Common Link (CL), which can be used by any TC. Having two separated sections, the framework guarantees a minimum bandwidth for any trunk and offers a common bandwidth which can be used by every trunk, irrespective to their TC. This allows for better bandwidth utilization and for the decrease of the rejected flows number. This paper is organized in the following manner. Section II presents related work, Section III describes the architecture and the functioning of the proposed framework, Section IV and Section V present the admission control method, and respectively the self adaptive reconfiguration technique of the proposed framework and, finally, Section VI presents the experimental results and the concluding remarks.

2 Related Work

Integrated Services (IntServ) framework uses Resource Reservation Protocol (RSVP) to reserve bandwidth for each flow at every router along the path of the flows. Using per-flow based hop-by-hop signaling, consisting of PATH and RESV messages, Integrated Services provides end-to-end guarantees. These guarantees come with the overhead of processing per-flow bandwidth reservation and maintaining per-flow state at each router along the flow's path. Because this overhead is significant and is increasing along with the network size, IntServ presents scalability problems.

Differentiated Services (DiffServ) framework classifies packets into traffic classes at the boundary of the network. During the classification process each packet is marked according to its traffic class. The routers inside the network recognize the traffic class of the packets and, using a scheduling mechanism, forward each packet function of the per-hop behavior associated to the traffic class of the packet. In the case of this framework, the service is provided on a per-class basis instead of a per-flow basis as in IntServ framework. This approach removes the overhead specific to IntServ framework reason for which DiffServ framework is much more scalable. However, DiffServ framework does not exercise admission control at the edge of the network, so the network can be overloaded, reason for which this framework does not provide end-to-end guarantees.

On-Demand QoS Path (ODP) provides end-to-end QoS guarantees to individual flows with less overhead than in the case of IntServ, maintaining a similar scalability to the one of the DiffServ. Two types of routers are defined in this framework: edge and core. ODP exercises per-flow admission control and end-to-end bandwidth reservation at the edge of the network. Inside the network ODP differentiates the traffic classes as in the DiffServ. ODP organizes link bandwidth hierarchically. Each physical link is statically divided into several Provisioned Links (PLs), each PL being dedicated to a traffic class. Each PL is divided into several trunks, each trunk being dedicated to an edge router. An edge router keeps track of available bandwidth of its trunks and performs admission control locally without hop-by-hop signaling through network. The main disadvantage of ODP is that the bandwidth adjustment is only inside the traffic class and does not allow for bandwidth redistribution between classes. The free bandwidth of the Provisioned Links that are not used or present a low utilization can not be made available for other Provisioned Links, the free bandwidth remaining unused. Another disadvantage of this framework is the fact that it does not include a module for determination of the necessary bandwidth for each input flow.

3 The Architecture of the Framework

The proposed framework serves the user networks and defines two types of routers, edge and core, and entities for common bandwidth control.

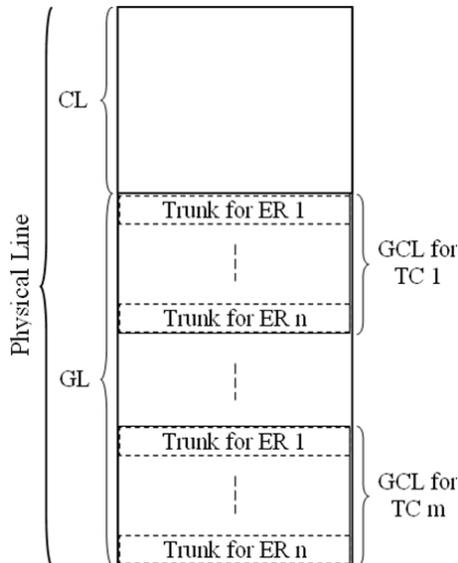


Figure 1: Bandwidth organization in the proposed framework

Edge routers (ERs), connected to served networks, determine the necessary bandwidth for each input flow, take admission or rejection decision for each input flow, dynamically reconfigure the bandwidth assigned to trunks, map flows to corresponding TCs and transmit the packets belonging to the admitted flows in the network. Core routers (CRs), connected to edge or core routers, recognize TCs and provide class based service differentiation. Entities for common bandwidth control monitor and update common bandwidths utilization and accept or reject the requests for additional bandwidth for trunks received from ERs.

The bandwidth is hierarchically organized. Each Physical Line is divided in two sections as it is presented in Figure 1. A first section guarantees the minimum bandwidth, which can be also 0, for each class and each trunk. The second section, CL, offers a common bandwidth which can be used by every trunk function of their bandwidth requirements, irrespective to their belonging TC or ER. So, trunks can acquire additional bandwidth without being conditioned by the available bandwidth of the belonging class. First section is statically divided in several Guaranteed Class Links (GCLs). Each GCL is reserved to a TC existing a one to one mapping between the TCs supported by the Physical Line and GCLs. Each GCL is divided in several trunks, each trunk being dedicated to an ER. A trunk belonging to a GCL supports the flows belonging to the TC that corresponds to the considered GCL, originating from the ER to which the trunk is dedicated, irrespective to their destination. An ER keeps track of available bandwidth of its assigned trunks and performs admission control locally, without hop-by-hop signaling through network. A Virtual IP Path (VIP) is a path from a source ER to a destination ER for a TC, being a concatenation of trunks belonging to the source ER over a source-destination path.

The bandwidth assigned to trunks has a minimum guaranteed value which can be also 0 and, by using CL, is dynamically adjusted function of the network traffic modifications.

Function of the entities for common bandwidth control there are three possible approaches: Central Control (CC), Router-Aided (RA) and Edge-to-Edge (EE).

The architecture of the framework is presented in Figure 2 and it is composed of two en-

tities: edge router and entity for common bandwidth control. The edge router determines the necessary bandwidth for each input flow, takes the admission or rejection decision for each input flow, reserves the necessary bandwidth for each admitted flow, dynamically reconfigures the bandwidth assigned to trunks and classifies the packets belonging to the admitted flows. The entity for common bandwidth control monitors and updates common bandwidths utilization and accepts or rejects the additional bandwidth requests for trunks, received from edge routers. The communication between the two entities is realized through a predefined message set.

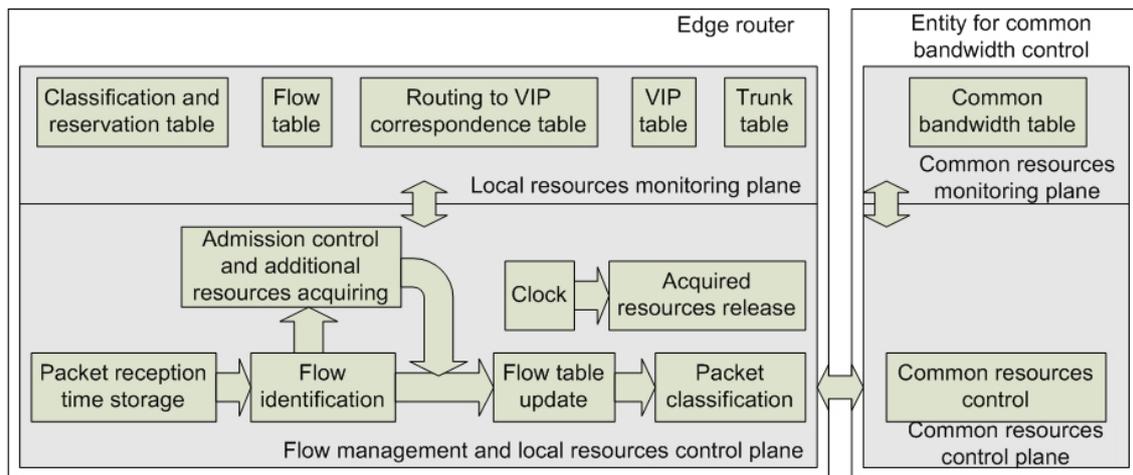


Figure 2: The architecture of the proposed framework

The edge router is composed of two planes: local resources monitoring plane and flow management and local resources control plane. The local resources monitoring plane is composed of the following tables: classification and reservation table which realizes a correspondence between flow types, the elements that identify them, corresponding traffic class, their necessary bandwidth and the maximum necessary bandwidth for any flow from the respective traffic class, flow table which stores the admitted flows and the time of the last packet from each flow, routing to VIP correspondence table which allows for VIPs determination, VIP table which stores the VIPs and trunk table which stores the reserved bandwidth, bandwidth being used and minimum reserved bandwidth for every trunk belonging to the ER. The flow management and local resources control plane takes the packets from the traffic policy module and delivers them to the routing process being composed of the following blocks: packet reception time storage which reads the receiving time of each packet, flow identification which determines and identifies the packets membership to admitted flows, flow table update which updates the reception time of the last packet from each flow from flow table, admission control and additional resources acquiring which admits the flows for which there are enough resources and rejects the flows when there is not enough bandwidth for them, acquires additional bandwidth for trunks, reserves the necessary bandwidth for the admitted flows and inserts the admitted flows in the flow table and packet classification which identifies the packets function of classification and reservation table criteria and marks them according to the identification criteria. The second task of this plane is to determine finished admitted flows and release the acquired resources used for these flows. The following blocks realize this task: clock generates the time period when acquired resources are released and acquired resources release which determines finished admitted flows and releases reserved acquired resources for these flows.

The entity for common bandwidth control is composed of two planes: common resources monitoring plane and common resources control plane. The common resources monitoring plane

contains common bandwidth table which stores the reservation and utilization for common bandwidths of CLs. The common resources control plane contains the common resources control block which updates the common bandwidth table and decides if additional bandwidth requests for trunks received from ERs can be accepted or not.

4 Admission Control

Admission control is performed at the arrival of the first packet from a new flow, by the source ER. Admission control and additional resources acquiring module stores the packet into the not admitted flows memory and determines if there are other packets belonging to this flow stored in the memory. If there are no more such packets, it determines from classification and reservation table the necessary bandwidth for the flow and TC, determines from routing to VIP correspondence table the flows corresponding VIP and extracts from VIP table the trunks that belong to the determined VIP. Then, for trunks which have enough available bandwidth, reserves the flows necessary bandwidth by updating the Bdw being used field. For a trunk, the condition to have enough available bandwidth is:

$$\text{Reserved_Bdw} \geq \text{Bdw_being_used} + \text{Necessary_Bdw} \quad (4.1)$$

where Reserved_Bdw and Bdw_being_used are the amounts of reserved and utilized bandwidth for the trunk and Necessary_Bdw is the flows necessary bandwidth.

The update of the Bdw_being_used field is done in the following manner:

$$\text{Bdw_being_used} = \text{Bdw_being_used} + \text{Necessary_Bdw} \quad (4.2)$$

If the VIP has enough bandwidth to support the input flow, the admission control accepts the flow. If there are trunks which do not have enough available bandwidth, admission control and additional resources acquiring module tries to increase the reserved bandwidth of those trunks sending in this sense a request to the entities for common bandwidth control. If the request is admitted, the reserved bandwidth of the trunks is increased by updating Reserved_Bdw field, so that these trunks too will have enough available bandwidth to support the input flow. For these trunks, admission control and additional resources acquiring module reserves the flows necessary bandwidth by updating the Bdw_being_used field. In this case too, the admission control accepts the flow. After a flow acceptance, the flow is inserted into the flow table and the packets belonging to this flow, stored into the not admitted flow memory, will be transmitted to the flow table update module for the rest of the processing and transmission. If the request is rejected, the flow is rejected, the reservations made on the trunks which had enough available bandwidth are canceled by updating the Bdw_being_used field and the packets belonging to the flow, stored into the not admitted flow memory, are discarded.

A flow is considered finished after an inactivity period that exceeds a predefined value. Each ER, using the acquired resources release module, periodically inspects its own flow table in order to identify the finished flows and, as a consequence of finished flow identification, releases the bandwidths correspondingly. If there are finished flows, the reserved bandwidth and TC for these flows are determined from the classification and reservation table and the flows are discarded from the flow table. Then, the acquired resources release module, determines from routing to VIP correspondence table the corresponding VIPs and extracts from the VIP table the trunks belonging to the determined VIPs. After this, releases the reserved bandwidth for the flows by updating the bandwidth being used field from the trunk table for each trunk belonging to the VIPs. The update of the Bdw_being_used field is done in the following manner:

$$\text{Bdw_being_used} = \text{Bdw_being_used} - \text{Necessary_Bdw} \quad (4.3)$$

Also, it extracts from the classification and reservation table the maximum amount of bandwidth for the corresponding TCs and verifies if the trunks utilization is under the predetermined lower threshold. For a trunk, the condition to have the utilization under a predetermined lower threshold is:

$$\text{Reserved_Bdw} > \text{Bdw_being_used} + n * \text{TC_maximum_necessary_Bdw} \quad (4.4)$$

where $\text{TC_maximum_necessary_Bdw}$ is the maximum amount of bandwidth for the corresponding TC and n is a predefined parameter having a value larger than or equal to 1.

Also, it extracts from the trunk table the minimum reserved bandwidth for the trunks and verifies if the trunks have additional bandwidth acquired from CLs. For a trunk, the condition to have additional bandwidth acquired from CL is:

$$\text{Reserved_Bdw} > \text{Trunk_minimum_reserved_Bdw} \quad (4.5)$$

where $\text{Trunk_minimum_reserved_Bdw}$ is the minimum reserved bandwidth for the trunk

If there are trunks whose bandwidth being used is under the predetermined lower threshold and the trunks have additional bandwidth acquired from CLs, the acquired resources release module, in the limit of the acquired bandwidth, computes de bandwidth that will be released from the reserved bandwidth of the trunks. Reduction of the reserved bandwidths is accompanied by appropriated resources release for the common bandwidths.

5 Self-Adaptive Bandwidth Reconfiguration

The proposed framework dynamically adjusts the bandwidth assigned to the trunks, in order to adapt to changes in network traffic. A source edge router has the option to request additional bandwidth for its trunks or it can release bandwidth not used by the trunks, depending on bandwidth usage of his trunks. Bandwidth adjustment is done using the CL's bandwidth. This adjustment allows all trunks, regardless of the class of traffic or the edge router where they belong, to share the bandwidth provided by LC. The trunk reconfiguration process of the proposed framework involves three main actions: (1) the control of the Common Bandwidth Table, (2) the release of bandwidth not used by the trunks, and (3) acquisition of additional bandwidth for trunks.

A Common Bandwidth Table stores the common bandwidth utilization of the network CLs. As shown in Figure 3, an entry in this table contains: CLs identifier, the reserved amount of shared bandwidth and the amount of shared bandwidth for the CL.

| <i>CL ID</i> | <i>Reserved BdwC</i> | <i>BdwC being used</i> |
|--------------|----------------------|------------------------|
| | | |

Figure 3: Common Bandwidth Table

Depending on the share bandwidth entities, three approaches are being proposed: Central Control (CC), Router-Aided (RA) and Edge-to-Edge (EE). In the Central Control approach the Common Bandwidth Table is managed by a network management server (NMS) and the Common Bandwidth Table stores the bandwidth utilization of all CLs in the network. In the Router-Aided approach, each core router manages a Common Bandwidth Table, and each of these tables stores the bandwidth utilization of the LCs belonging to all physical links directly connected to that core router. In the Edge-to-Edge approach each edge router manages a Common Bandwidth Table, which will store the bandwidth utilization of all the CLs in the network.

1. If $Reserved_Bdw > Bdw_being_used + n \times TC_maximum_necessary_Bdw$ and $Reserved_Bdw > Trunk_minimum_reserved_Bdw$
 - 1.1. $Released_Bdw = Reserved_Bdw - Bdw_being_used - n \times TC_maximum_necessary_Bdw$
 - 1.2. If $Trunk_minimum_reserved_Bdw > Reserved_Bdw - Released_Bdw$
 - 1.2.1. $Released_Bdw = Reserved_Bdw - Trunk_minimum_reserved_Bdw$
 - 1.3. $Reserved_Bdw = Reserved_Bdw - Released_Bdw$

Figure 4: Reserved bandwidth update algorithm

Each edge router periodically examines its own Flow Table and determines which flows are finished. If there are any finished flows, the flow table and the trunk table will be updated. A next step for the edge router is to examine the trunk table and to obtain the bandwidth utilization of its own trunks. If the bandwidth utilization of any trunk is under a predetermined lower threshold and those trunks have additional bandwidth acquired from the Common Link, the source edge router computes the amount of bandwidth to be released from the reserved bandwidth, adjusts the released bandwidth of the trunks, in the limit of the additional acquired bandwidth, updates its own trunk table and sends a control message to the entities for common bandwidth control, in order to release the used shared bandwidth. Adjusting a trunk's bandwidth is done only in the limit of the additional acquired bandwidth. The algorithm that describes the reserved bandwidth update process for the trunks is presented in figure 4.

The trunk reconfiguration process is always initiated by a source edge router using a threshold and computed values driven mechanism.

6 Experimental Results and Conclusions

For the development and testing of the proposed QoS framework (SAR framework) and also for the developing of new ones, an experimental methodology was used, rather than simulation techniques, thus an integrated solution - a development tool, was created [3]. Also a benchmarking system for QoS parameters [4] was developed in order to allow the testing of the proposed SAR QoS framework. The benchmarking system generates traffic for the defined testbed and measures the following parameters: delay, IP delay variation (IPDV) or jitter and bandwidth, both on TCP and UDP. The benchmarking allows a user to define and store complex traffic patterns that can be recharged for making further measurements, to test various QoS techniques based on the same traffic characteristics.

For simulations purposes, the Self-Adaptive bandwidth Reconfiguration QoS framework (SAR) described in the previous section and the ODP framework were tested, in a comparative manner, using the development tool and the benchmarking system. The final testbed is a network of programmable routers, and consisted of three edge routers and three served networks. The tests were intended as performance comparison between ODP and SAR frameworks. Traffic classes and traffic patterns were defined similarly in both frameworks tested. Four classes of traffic were considered. Two test traffic patterns were defined. In the first traffic pattern considered flows are injected from classes 2 and 3 and in the case of the second traffic pattern flows belonging to class 2 are injected. For both traffic patterns a balanced distribution of traffic from and to the served networks is ensured.

After testing and analyzing the results (Figure 5) it was found that the number of flows

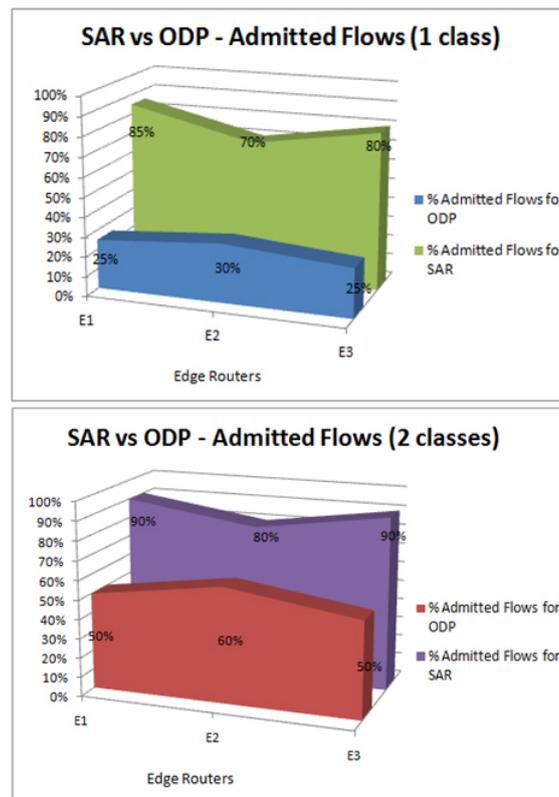


Figure 5: Test Results - Admitted Flows

admitted for SAR framework is higher than in the case of ODP framework, on both tested traffic patterns, which demonstrates a more efficient use of network resources. Also, the equal number of control messages transmitted by the two frameworks shows that SAR is a scalable framework. Finally, tests confirmed that admission control has eliminated network congestion.

This paper presents a new end-to-end QoS framework, called Self-Adaptive bandwidth Reconfiguration QoS framework (SAR). The proposed dynamic allocation method guarantees a minimum bandwidth available for each traffic class and trunk, and provides a common bandwidth section which can be used by every trunk, function of their bandwidth requirements, irrespective to their belonging TC or ER. Thus, trunks can acquire additional bandwidth without being conditioned by the available bandwidth of the belonging class. The new framework, SAR, uses the proposed bandwidth organization, allowing the increase of the traffic volume it handles, guaranteeing end-to-end quality of service through network resources monitoring, admission control and resource reservation for new flows. The end-to-end QoS framework with self-adaptive bandwidth reconfiguration overcomes the disadvantages of ODP by providing minimum service guarantees and bandwidth redistribution between classes.

Acknowledgments

This work was supported by the PNII-IDEI 328/2007 QAF - Quality of Service Aware Frameworks for Networks and Middleware research project within the framework - National Research, Development and Innovation Programme initiated by The National University Research Council - Romania (CNCSIS - UEFISCSU).

Bibliography

- [1] Z. Wang, *Internet QoS: Architectures and Mechanisms for Quality of Service*, Morgan Kaufmann, San Francisco, 2001.
- [2] M. Yang, Y. Huang, J. Kim, M. Lee, T. Suda, M. Daisuke, An End-to-End QoS Framework With On-Demand Bandwidth Reconfiguration, *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Hong-Kong, Vol. 3, pp. 2072 - 2083, 2004.
- [3] A. Peculea, V. Dadarlat, I. Ignat, B.Iancu, L. Cobarzan, On Developing a Qos Framework With Self-Adaptive Bandwidth Reconfiguration, *Pollack Periodica An International Journal for Engineering and Information Sciences*, Vol.4, No.1, pp. 121-129, 2009.
- [4] A. Peculea, B. Iancu, V. Dadarlat, I. Ignat, E. Cebuc, Z. Baruch, Benchmarking System for QoS Parameters, *Proceedings of the IEEE 3rd International Conference on Intelligent Computer Communication and Processing 2007 (ICCP 2007)*, Cluj-Napoca, Romania, p.255-p.258, 2007.