Quality of Service Management in IP networks

M. Louta, A. Michalas, V. Loumos
School of Electrical and Computer Science Engineering
National Technical University of Athens
Athens, Greece
louta@telecom.ece.ntua.gr

E. Loutas
Department of Informatics
University of Thessaloniki
Thessaloniki, Greece
eloutas@zeus.csd.auth.gr

Abstract— **The DiffServ architecture provides a scalable mechanism for QoS introduction in an IP network. The idea of DiffServ is based on the aggregation of traffic flows at an ingress (or egress) point of a network and IP packet marking for different priority flows, according to several classification criteria. In this paper the problem of the improvement and fairness of absolute QoS provisioning to paths established along a DiffServ network on a per router basis is considered. The service rate reconfiguration problem of a router's output link is formally defined, mathematically formulated and solved by means of efficient heuristic algorithms, providing good solutions in reasonable time. Finally, an indicative set of results is provided and concluding remarks are made.**

Keywords- Quality of Service, Differentiated Services, Service Rate Reconfiguration, 0-1 Linear Programming.

I. INTRODUCTION

Service differentiation is considered to be of outmost importance for QoS provisioning in IP networks, due to the high variations of the connection requirements posed by Internet users and the statistical in general nature of the generated traffic, which the last years is presenting an exponential increase. The research community has concentrated on two different techniques to provide QoS differentiation to customers of packet switched networks. First, the Integrated Services (IntServ) [1][2] and the Differentiated Services (DiffServ) [3][4][5] approach. The major difference between IntServ and Diffserv architecture is the granularity of service differentiation. The IntServ concept lies in resource reservation notion, while in DiffServ model IP traffic is classified into finite, predefined service classes (on the basis of the demand requirements and characteristics) that receive different routing treatment. DiffServ achieves scalability and manageability by providing quality per traffic aggregate and not per application flow, while on the other hand IntServ approach faces potential problems.

Two directions exist in the DiffServ architecture, the relative and the absolute. In absolute DiffServ [6] architecture, strict QoS parameters are defined for each service class. An admission control scheme is used [7] to provide QoS guarantees as absolute bounds of specific QoS parameters such as bandwidth, packet transfer delay, packet loss rate, or packet delay variation (jitter). For any accepted user the appropriate resources are reserved and the level of performance of his connection is assured. The relative DiffServ model [8] provides QoS guarantees per class in reference to guarantees given to other classes. The only assurance from the network is that higher classes receive better service treatment than lower classes. Proposals for relative per class DiffServ QoS define service differentiation qualitatively [9][10] in terms that higher classes receive lower delays and losses from lower classes. Relative service differentiation is a simple and easy deployed approach compared to the absolute service differentiation.

Taking into account the fact that most admission control schemes consider average traffic arrival rate, in conjunction with the non static in general source's behaviour, congestion is likely to emerge on the core network routers. For example, let us consider the case where one (or more) service classes utilise their allocated portion of bandwidth. Thus, an incremental differentiation to a source traffic arrival rate contributing to this (those) service class(es), exceeds the bandwidth reserved and thus, the extra generated traffic cannot be accommodated. In such a case, reconfiguration of routers is needed in order to provide the best QoS possible per flow. Consequently, assuming that admitted users and applications cannot get the requested absolute service level assurance, such as an end-to-end delay bound or throughput due to network resources insufficiency, a consistent service differentiation on output links of core routers can be provided, so that most of the QoS required levels are satisfied.

In the context of this paper, the problem of the improvement and fairness of absolute QoS provisioning to paths established along a DiffServ network on a per router basis through real-time service rate reconfiguration is considered. In this version, service differentiation is defined in terms of local packet delays per service class on each output link of a router. This consideration is based on the fact that a path's delay for packets of service class k, which covers from an ingress to an egress node, is bounded by $D_{pi,k}$. This parameter in essence is the sum of the worst case local delays suffered by service class k packets at each router's output link along the pi path.

The aim of this paper is to address the aforementioned problem from one of the possible theoretical perspectives. Specifically, the service rate reconfiguration problem of a router's output link is formally defined, mathematically formulated and solved by means of efficient heuristic

algorithms, providing good solutions in reasonable time. The version of the problem that is addressed in this paper exploits the available resources of service classes in order to improve QoS characteristics of the problematic flows. A possible extension to this version is the accomplishment of the strict QoS constraints of the higher order classes by relaxing the QoS constraints of the lower service classes. Specifically, service classes rated less than the one(s) presenting the problem at a specific time instance during its(their) lifetime, will give part of its (their) resources in order to accommodate traffic of the higher rated service classes.

The paper is organised as follows. In section II, a formal model of the service rate reconfiguration problem is concisely defined. Section III presents a mathematical formulation of the service rate reconfiguration problem, while in Section IV, the problem is solved by means of an heuristic algorithm. As a first phase the service rate reconfiguration problem attempts to meet, for each service class, its absolute QoS constraints. As a second phase, in case the absolute QoS characteristics have not been met, the strict constraints of the lower service classes are relaxed, in order to gain the absolute QoS of the higher rated service class, which in essence covers the extended version of the problem. In Section V, a set of results, indicative of the performance of our proposed reconfiguration scheme is presented.

II. FORMAL PROBLEM STATEMENT

In the framework of the Absolute DiffServ Architecture, we assume the existence of N service classes. The set of service classes is denoted by SC and for each service class i $(i \in SC)$, the service rate assigned to is $sr(i)$. It holds that $\sum_{i=1}^{N} sr(i) = B$, where B is the bandwidth budget of the link. The QoS characteristic considered in the context of this paper is the average packet delay for service class i $(i \in SC)$. Let's, $d_g(i)$ denote the absolute QoS constraint regarding queuing delay of service class i packets.

Each service class i is associated with a relative importance factor, denoted as $RF(i)$, that in essence rates service class i according to the serving priority attributed to it. A fundamental assumption at this point is that the importance factor $RF(i)$ for service class i may be defined by network operators, along with the link sharing hierarchy, the amount of bandwidth assigned to each service class and the absolute QoS parameters. In our study, parameter $RF(i)$ assumes higher values for higher order classes, as we consider of outmost importance to best serve these service classes with respect to the lower rated ones. Furthermore, we assume the existence of a monitoring module, which informs us in case of QoS violation for service class i $(i \in SC)$. P denotes the set of service classes that at the time instance t present a delay related QoS violation. Assuming that $d(i,t)$ represents the packet delay of service class i at time t, the following equation holds $P = \{i \in SC \mid d(i,t) > d_g(i)\}$.

In the specific framework considered, the effect of any delay related QoS violation is minimised by utilising service class j $(j \in SC)$ resources. In essence, a *service rate reconfiguration scheme* will be adopted in order to succeed in bringing the best possible QoS for the service classes. In this study it is assumed that all service classes are bounded by strict absolute QoS. Thus, the delay related QoS constraint of the service classes not presenting a violation at time instance t should be preserved after the application of the reconfiguration scheme, while the QoS characteristics of the rest service classes should be improved. This model may readily integrate notions from the Relative approach as well, in order to satisfy the strict absolute constraints of the higher order classes by relaxing the strict constraints of the lower rated ones.

Let's C represent the set of candidate service classes for allocating a portion of their resources to service classes belonging to P set. In the confined version of the service rate reconfiguration problem, the set C may be constituted by service classes, which at time instance t comprise available resources, that could be assigned to service class $j, (j \in P)$. Thus, $C = \{i \in SC \mid d(i,t) < d_g(i)\}$, $\forall j \in P$. In the extended version of the service rate reconfiguration problem, for each service class j $(j \in P)$, the set of candidate service classes comprises additionally, all the lower rated service classes i $(i \in SC)$ with respect to service class j. Thus, $C_j = \{i \in SC \mid d(i,t) < d_g(t)\} \cup \{i \in SC \mid RF(i) < RF(j)\}$, $j \in P$.

Service class i, $i \in SC$ may in general be associated with two cost factors $cf_u(i)$ and $cf_r(i)$, expressing the cost of providing part of available and not available resources to a service class j, $j \in P$. In essence, the second cost factor $cf_r(i)$ is introduced in the extended version of the service rate reconfiguration problem. These cost factors may be related to specific characteristics of service class i (i.e., importance factor $RF(i)$, historical data regarding QoS service provisioning problems that service class i experienced in the past). In our study, for the determination of these cost factors, the importance factor $RF(i)$ of service class i is taken into account. Specifically, for lower rated classes, the cost factors are considered to impose an insignificant burden, since higher rated service classes are attributed with higher serving priority. Moreover, the cost of utilising available resources is considered trivial to the cost of relaxing the absolute QoS constraints of a service class. Let's A_j denote the set of service classes that assign portion of their resources to service class j $(j \in P)$ and $p(i,j)$, $0 \le p(i,j) \le 1$ denote the service rate portion that service class i allocates to service class j $(i \in C, j \in P)$. Therefore, $A_j = \{(i,p) \mid i \in C(C_j) \,\&\, p(i,j) > 0\}$, $j \in P$.

The objective of the service rate reconfiguration problem, in case of delay related QoS violation, is to find a new service rate configuration $SRC(t) = \{sr(i) \mid i \in SC\}$ on the basis of the aforementioned allocation $A = \{A_j \mid j \in P\}$, i.e., an

configuration of service rates $sr(i)$ to service classes $i \in SC$, which should maximise a cost function $f(SRC(t))$, that is associated with the overall packet delay related QoS characteristics of service classes at time instance t. Among the terms of this function there can be the overall anticipated QoS improvement with respect to the problematic service classes that results from the new service rate configuration and which may be expressed by the function $b_c(SRC(t))$ and the cost associated with the provision of the new service rate configuration which is expressed by the function $c_c(SRC(t))$.

The constraints of our problem are the following. First, all resources of each service class should be allocated to itself and to service classes j, $(j \in P)$. Therefore, $\sum_{j \in P \cup \{i\}} p(i, j) = 1, \ \forall i \in C(C_j)$. Second, the delay constraints of each of the non problematic service classes should be preserved. More specifically, service class i, $(i \in C(C_j))$ should not allocate more resources than required in order to satisfy its own delay related QoS constraints. In case the extended version of the service rate reconfiguration problem is considered, this constraint is relaxed in order to succeed in meeting the strict absolute QoS constraints of the higher order service classes with respect to the lower order classes. In such a case, the requirement posed to each service class is to have a minimum service rate. Thus, $sr(i) \neq 0, \ \forall \mathrm{i} \in \mathrm{SC}$.

The version of overall problem can be formally stated as follows.

[Service Rate Reconfiguration Problem]

Given: (a) the set of service classes SC, (b) the set of candidate service classes C for allocating portion of their resources, (c) the importance factor $RF(i)$ and the cost factor $cf_u(i)$ associated with each service class i, (d) the service rate $sr_{pre}(i)$ already attributed to service class i at time instance t, (e) the absolute delay related QoS constraint $d_g(i)$ for each service class i, (f) the packet delay $d_{pre}(i)$ encountered by service class i at time instance t, (g) the load of the queue of service class i in bits $q(i)$, find the best service rate configuration pattern associated with the allocation $A = \{A_j \mid j \in P\}$, i.e., assignment of service rates to service classes $SRC(t)$, that optimises an objective function $f(SRC(t))$ that is related to the overall anticipated QoS improvement $b_c(SRC(t))$ with respect to the problematic service classes which results from the new service rate configuration and the cost $c_c(SRC(t))$ associated with the provision of the new service rate configuration, under the constraints $d_{post}(i,t) \leq d_g(i), \ \forall i \in C$, where $d_{post}(i)$ is the packet delay service class i is encountering after the service rate reconfiguration and that all resources should be assigned to a service class.

The above general problem is open to various solution methods. Its generality partly lies in the fact that the objective and the constraint functions are open to alternate implementations. The problem statement can be distinguished from the specific solution approach adopted in the next subsection. At this point, it should be noted that the extended version of overall problem may readily be provided by introducing the extensions described above.

III. Optimal Formulation

In this sub-section the problems above are formulated as a 0-1 linear programming [11][12]. The experimentation and comparison with important alternate formulation approaches is a stand-alone issue for future study. In order to describe the configuration $SRC(t)$ of service rates to service classes, the decision variables $x(i, j, p)$ $(i \in C(C_j), \ j \in P, \ p \in [0,1])$, which take the value 1(0) depending on whether the service class-i assigns (does not assign) to service class-j p portion of its resources, are introduced. Additionally, the decision variables $\psi(i)$ assume the value 1(0), depending on whether service class-i presents (does not present) a delay related QoS violation. Thus, $\psi(i) = 1(0)$ if $i \in (\notin) P$. The problem of obtaining the most appropriate configuration $SRC(t)$ according to the confined version may be obtained by reduction to the following optimisation problem.

[Service Rate Reconfiguration Problem]

Maximise:

$$f(t, SRC(t)) = \sum_{i \in SC} [d_{pre}(i) - d_{post}(i)] \cdot RF(i) \cdot \psi(i) - \sum_{i \in SC} [d_{post}(i) - d_{pre}(i)] \cdot cf_u(i) \cdot (1 - \psi(i)), \ \forall i \in SC \quad (1)$$

where $d_{post}(i)$, $d_{pre}(i)$ is the packet delay encountered by service class $i \in SC$ at time instance t, after/before applying the service rate reconfiguration scheme, respectively. In essence, the $d_{post}(i)$, $d_{pre}(i)$ parameters are dependent on the load of the queue of service class i at time instance t and the service rate allocated to the service class i, $sr_{post}(i)$ and $sr_{pre}(i)$, respectively. Specifically, considering a packet arriving at time instance t, it is serviced after the current queue load $q(i)$ has been serviced. Thus, the following equation holds:

$$d_{post,pre}(i) = \frac{q(i)}{sr_{post,pre}(i)}, \ \forall i \in SC \quad (2)$$

subject to:

$$\sum_{j \in P \cup \{i\}} p \cdot x(i, j, p) = 1, \ \forall i \in C \quad (3)$$

$$d_{post}(i,t) \le d_g(i),\ \forall i \in C \quad (4)$$

$$SRC(t) = \{sr_{post}(i) \mid i \in SC\} \quad (5)$$

For the parameter $sr_{post}(i)$, the following equation holds:

$$sr_{post}(i) = \begin{cases} sr_{pre}(i) \cdot [1 - \sum_{j \in P} p \cdot x(i,j,p)],\ \psi(i) = 0 \\ sr_{pre}(i) + \sum_{j \in C} p \cdot x(j,i,p) \cdot sr_{pre}(j),\ \psi(i) = 1 \end{cases},$$
$$\forall i \in SC \quad (6)$$

$$b_c(SRC(t)) = \sum_{i \in SC} [d_{pre}(i) - d_{post}(i)] \cdot RF(i) \cdot \psi(i) \quad (7)$$

$$c_c(SRC(t)) = \sum_{i \in SC} [d_{post}(i) - d_{pre}(i)] \cdot cf_u(i) \cdot (1 - \psi(i)) \quad (8)$$

Relation (1) expresses the objective of finding the best assignment of service rates to service classes that minimises the cost function, which is associated with the overall delay related QoS characteristics of the service classes. In other words, relation (1) expresses the satisfaction stemming from the improvement of the quality of the service classes belonging in the set P, $b_c(SRC(t))$-relation (7), as well as the cost $c_c(SRC(t))$-relation (8), stemming from the deterioration of the quality of the service classes that assign portion of their resources to other service classes.

Relations (5) and (6) express the new configuration scheme regarding service rates for each service class i $(i \in SC)$.

Constraints (3) guarantee that each service class i $(i \in C)$ will assign all its resources. Constraint (4) guarantees that the delay related QoS characteristic of the service classes i, $i \in C$ will not present any violation after the reconfiguration, thus their absolute QoS requirements will be met.

IV. Computationally Efficient Solutions

This section discusses computationally efficient solutions for the problem of service rate reconfiguration that is addressed in this paper. In general, there may be a significant amount of computations associated with the optimal solution of problems 1 and 2. In this respect, the design of computationally efficient algorithms that may provide good (near-optimal) solutions in reasonable time is required. Classical methods in this respect are simulated annealing [13], taboo search [14], genetic algorithms [15], greedy algorithms etc. Hybrid or user defined heuristic techniques may also be devised.

In case the size of the problem instance (the service classes presenting delay related QoS violation and the candidate service classes for assigning part of their resources) is not prohibitively large, a solution method can be to exhaustively search the solution space. Otherwise, the response time of the exhaustive search is impractical and the reconfiguration of the network cannot be done quickly and efficiently.

The algorithm adopted in this paper for the solution of the *service rate reconfiguration problem* (both the confined and the extended version) follows the simulated annealing paradigm.

A. Algorithm based on Simulated Annealing

Such algorithms are required in case the solution space is prohibitively large to be scanned in an exhaustive manner. During each phase of an algorithm that is based on the simulated annealing paradigm, a new solution is generated by minimally altering the currently best solution (in other words, the new solution is chosen among those that are "neighbouring" to the currently best one). If the new solution improves the objective function value (i.e., the difference between the objective function value of the old and the new solution, Δc, is negative) the new solution becomes the currently best solution. Solutions that decrease the objective function value may also be accepted with probability $e^{-(\Delta c/CT)}$ (Metropolis criterion). This is a mechanism that assists in escaping from local optima. CT is a control parameter, which may be perceived as the physical analogous of the temperature in the physical process. The algorithm ends when either $CT = 0$ (temperature reaches 0) or when a significant number of moves have been made without improving the cost function.

The development of a simulated annealing-based procedure means that the following aspects have to be addressed: configuration space, cost function "neighbourhood" structure and cooling schedule (i.e., manner in which the temperature will be reduced). The configuration space is the set of feasible solutions $x(i,j,p)$, where $\psi(j) = 1$, $(j \in P)$ and $\psi(i) = 0$, $(i \in C(C_j))$, that satisfy the constraints (2)-(8). The cost function is the one introduced by relation (1).

The neighbourhood structure of a solution is produced by reallocating portion of service class i resources $(i \in C(C_j))$ from its current service class j, $(j \in P)$ to another randomly chosen (higher or lower) service class j', $(j' \in P)$. The cooling schedule may be calculated according to $T' = r \cdot T$, where T is the temperature and r is usually a number that ranges from 0.95 to 0.99.

V. Results

In this section, some indicative results are provided in order to assess the proposed software framework, which allows for QoS characteristics improvement of the problematic flows in IP networks. The results of this section aim at the provision of indicative evidence of the efficiency and the effectiveness of the proposed service rate reconfiguration scheme. Even though the experiment conducted entails the simplest possible case, as

it comprises only one router, it enables the acquisition of an initial set of indicative results that show the behaviour of our scheme.

In a real network environment, the network administrator may be supported by a Configuration and Reconfiguration Management Subsystem. This system caters for the initial configuration of the routers (i.e., define parameters for the DiffServ service classes on the basis of past experience/ historical data). Additionally, it monitors the network load and QoS conditions and adjust the service rates in case the predefined thresholds are about to be exceeded, while in parallel a notification is send to the network operator. Thus, action is taken before the SLA is breached and the customer experience is adversely affected. For the implementation of this module, the technology of Mobile Intelligent Agents [16] may be adopted. Mobile intelligent agents performing the reconfiguration tasks should be sent from the management station as close as possible to the routers in order to complete their mission.

In order to test the performance of the proposed framework of this paper we used NS2 network simulator [17] developed by National Berkley Labs as the simulation platform, enhanced with a novel algorithm following the simulating annealing technique which realises the service rate reconfiguration scheme presented in this study. Fig. 1 shows the topology adopted for the realisation of the set of experiments. Four source nodes s_1, s_2, s_3, s_4 generate traffic to their destinations nodes d_1, d_2, d_3, d_4, respectively. Incoming packets are classified into 4 classes with *class-1* having the lowest priority and *class-4* the highest. Packets from s_1 to d_1 are classified as *class-1* packets, from s_2 to d_2 as *class-2* packets and so on. In the context of our study, we considered Pareto sources nodes with shape parameter a=1.9 and mean on and off time 5msec. This choice was driven by the fact that, even in aggregate, Pareto sources exhibit highly bursty characteristics. The packet length of incoming traffic is taken equal to 1Kbyte for all classes.

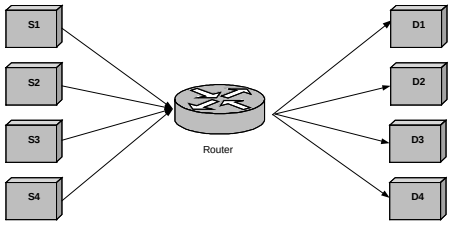


Figure 1. Simulation topology

Packets are passed from their sources to their destinations through a CBQ scheduler [18] with output link capacity of 10Mbps. CBQ is a link sharing scheduler, which is variation of the GPS algorithm [19]. It is based on several mechanisms that merge Priority Queueing (PQ) and fair capabilities to provide differentiated services to service classes. While CBQ internal mechanisms are quite complex, its use is quite simple. Network managers need to define the link-sharing hierarchy and assign the amount of bandwidth and priority of each class. Due to its intuitiveness, CBQ is considered to be the most appealing advanced scheduler available today for the differentiated service provisioning.

Table I presents the values of all input parameters considered in the experiment.

TABLE I. Input parameters

parameters \ *classes*	*class1*	*class2*	*class3*	*class4*
Generated Traffic(Mb)	2.138	4.256	6.363	8.3648
$RF(i)$	1	10	100	1000
$cf_u(i)$	0.0001	0.001	0.01	0.1
$sr_{pre}(i)$ (Mb)	1	2	3	4
$d_g(i)$ (msec)	350	280	150	80

In the reconfiguration experiments performed, the local delay experienced by each service class i, $d_{pre}(i)$, is calculated on the basis of (2) considering regular time intervals, hereby denoted as U. It is obvious that a small U would increase processing load in the routers. A large U would result in packet delay approximations per service class that will not conform to the real ones. As in [20], we find that a for 0.001sec $< U <$ 0.1sec the behavior of the scheme is good. In case that the predefined threshold values d_g are violated, suitable $sr_{post}(i)$ values are found using (1). Specifically, the $sr_{post}(i)$ values, are calculated so that the objective function (1) is maximized, while the absolute delay constraints $d_g(i)$ are satisfied for each service class. An algorithm based on simulated annealing technique has been realised for the provision of a good solution (near the optimal) within reasonable time.

The Service Rate Reconfiguration scheme may be adopted on static or on a dynamic base. In the static case, the service rates are adjusted to the CBQ scheduler once, when a reconfiguration request is issued by the network administrator. In such a case, the monitoring module of the Configuration Management Subsystem notifies network administrator in case of potential violation of the QoS parameters, thus the operator is enabled to take a corrective course of action. In the dynamic case, the service rates are dynamically adjusted when the predefined threshold values d_g are violated. At this point it should be noted that the minimum time between two successive service rate adjustments of the CBQ scheduler is U.

Fig. 2 illustrates the queuing packet delays per service class before the reconfiguration scheme has been applied. As it can be observed the fourth service class presents a QoS related violation, as most of the packets experience delays which are above their required $d_g(4)$ value that equals 80msec.

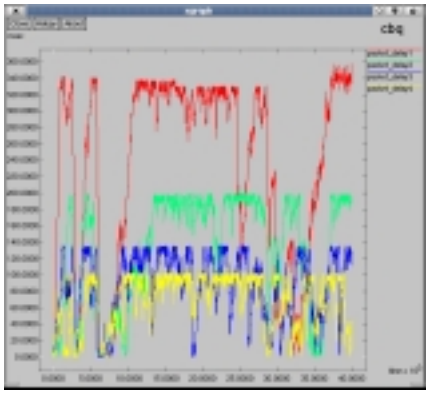
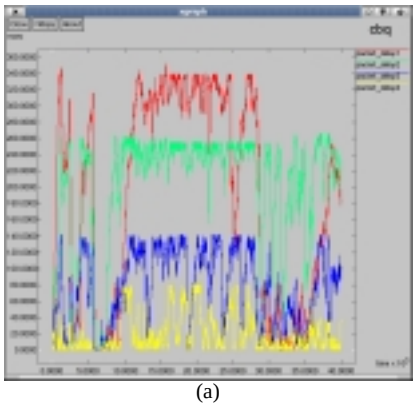
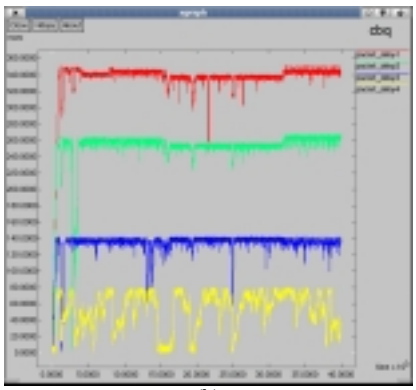


Figure 2. Individual packet delays prior to the reconfiguration

Both the static and dynamic cases have been considered in the experiment conducted. Fig. 3(a)-(b) present in a graphical manner the local delays experienced by the packets of each service class after the reconfiguration scheme, concerning the static and the dynamic case respectively. As it can be observed, in the static case the required absolute delay differentiation is satisfied for the average as well as for the individual packet delays per service class as long as the percentage of input traffic per class of service does not change. In the dynamic case, the reconfiguration scheme achieves the required absolute delay differentiation through the dynamic adaptation of service rates per service class. All service classes achieve average and individual packet delays, which are below the predefined threshold value $d_g(i)$.



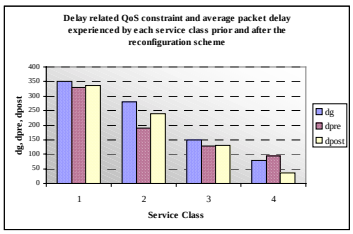
(a)



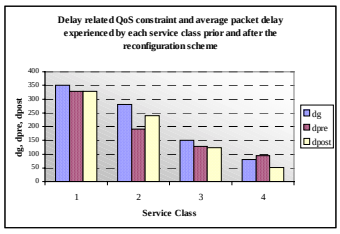
(b)

Figure 3. Individual packet delays after the reconfiguration scheme considering the static (a) and the dynamic case (b), respectively.

Considering the average packet delay each service class is experiencing prior to and after the application of the reconfiguration scheme, it may be noted that this parameter presents an overall improvement of approximately 28%. Fig. 4(a-b) depicts the delay related QoS constraint and the average packet delay experienced by each service class both for the static and the dynamic case.



(a)



(b)

Figure 4. Delay related QoS constraint and average packet delay experienced by each service class prior and after the reconfiguration scheme for the static and the dynamic case, respectively.

VI. CONCLUSIONS

In general, the scope of our paper is to augment the QoS management framework in IP networks. More specifically, the contribution of this paper lies in the definition, mathematical formulation and optimal as well as computationally efficient solution of the service rate reconfiguration problem that should be solved in the context of the IP network management. At the final sections the paper results are provided and concluding remarks are made.

Directions for future work include, but are not limited to the following. First, the realisation of further wide scale trials in a real network environment, so as to experiment with the applicability of the framework presented herewith. Second, the introduction of the extended version of the reconfiguration scheme in order to address the cases where the available resources of the service class prove to be inadequate for the satisfaction of the absolute delay constraints of each service class. Thus, in the extended version of the problem, the accomplishment of the strict QoS constraints of the higher order classes will be achieved by relaxing the strict QoS constraints of the service classes rated less than the one(s) presenting the problem.

REFERENCES

[1] P.P White, “RSVP and Integrated Services in the Internet: A Tutorial.” In *IEEE Communications Magazine*, May 1997.

[2] R. Braden, D. Clark, and S. Shenker, “Integrated services in the internet architecture: an overview”, IETF RFC1633, July 1994.

[3] Nichols, K., Blake, S., Baker, F. and D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, RFC 2474, December 1998.

[4] K. Nichols, V. Jacobson, and L. Zhang, “Two-bit differentiated services architecture for the Internet”, IETF RFC2638, July 1999.

[5] S. Blake, et al., “An Architecture for Differentiated Services”, RFC 2475, Dec. 1998.

[6] B. Teitelbaum, “QBone Architecture (v1.0)”, Internet2 QoS Working Group Draft, http://www.internet2.edu/qos/wg/papers/qbArch/1.0/draft-i2-qbone-arch-1.0.html, Aug. 1999.

[7] I. Stoika and H. Zhang, LIRA, “An approach for Service Differentiation in the Internet”, In *Proceedings NOSS-DAV*, 1998.

[8] C. Dovrolis, D. Stiliadis, “Relative Differentiated Services in the Internet: Issues and Mechanisms”, In *ACM SIGMETRICS Performance Evaluation Review* Vol. 27, No 1, June 1999.

[9] C. Dovrolis, D. Stiliadis, P. Ramanathan, “Proportional Differentiated Services: Delay Differentiation and Packet Scheduling”, In *IEEE/ACM Transactions in Networking*, Feb. 2002.

[10] C. Dovrolis, D. Stiliadis, Parmesh Ramanathan, “Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping”, In *Proceedings of the 2000 International Workshop on Quality of Service (IWQoS)*, Pittsburgh PA, June 2000.

[11] H. Salkin, “Integer programming”, Addison-Wesley, Reading, Massachusetts, 1975.

[12] C.Papadimitriou, K.Steiglitz, “Combinatorial optimization: Algorithms and complexity”, Prentice Hall, Inc.,1982

[13] E.Aarts, J.Korts, “Simulated annealing and the Boltzmann machines”, Wiley, New York, 1989

[14] F.Glover, E.Taillard, D. de Werra, “A User's Guide to Taboo Search”, In *Annals of Operations Research*, Vol. 41, 1993

[15] L.Davis, “Handbook of genetic algorithms”, Van Nostrand Reinhold, New York, 1991

[16] Bieszczad, A., White, T., and Pagurek, B., "Mobile Agents for Network Management", In *IEEE Communications Surveys*, Fourth Quarter 1998, Vol. 1, No. 1, September, 1998.

[17] S. McCanne and S. Floyd, http://www-mash.cs.berkeley.edu/ns/ns.html Network Simulator, 1996.

[18] S. Floyd and V. Jacobson, “Link-sharing and resource management models for packet networks”, In *IEEE/ACM Transactions on Networking*, 3(4):365–386, August 1995.

[19] H. Zhang, Service disciplines for guaranteed performance service in packet-switching networks, In *IEEE Proceedings*, vol. 83, pp. 1374-1399, Oct. 1995.

[20] A. Michalas, T. Kotsilieris, S. Kalogeropoulos, V. Fafali, G. Karetsos, V. Loumos and E. Kayafas, “Proportional Delay Differentiation Provision by Bandwidth Adaptation of Class Based Queue Scheduling”, Submitted to the *International Journal of Communication Systems*, Wiley Interscience.