Quality of Service Management in IP networks through Dynamic Service Rate Reconfiguration

M. D. Louta and A. C. Michalas

National Technical University of Athens, Division of Computer Science,
School of Electrical and Computer Engineering
9 Heroon Polytechneiou Str., GR-15773 Athens, Greece
tel: +3010 7721447 fax: +3010 7722534
e-mail: louta@telecom.ece.ntua.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 the IP packet marking for distinct 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. Specifically, the dynamic 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. Two versions of the problem are addressed. First, a confined version, which exploits the available resources of service classes in order to improve QoS characteristics of the problematic flows. Second, an extended version of the problem, in which the strict constraints of the lower classes are relaxed in order to strictly meet the absolute QoS requirements of the higher order service classes.

Key Words: Quality-of-Service, Differentiated Services, Admission Control, Dynamic Service Rate Reconfiguration, 0-1 Linear Programming.

1 INTRODUCTION

Service differentiation is considered to be of outmost importance for QoS provisioning in IP networks, required in order to effectively and efficiently support and manage the multitude of applications, the highly variable 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 for provision of 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 per application flow, while in DiffServ model IP traffic is classified into finite, predefined service classes with different levels of QoS (on the basis of the demand requirements and characteristics) that receive different routing treatment in accordance with the Per Hop Behaviors (PHBs) concept. DiffServ achieves scalability and manageability through QoS provision per traffic aggregate and not per application flow, while on the other hand IntServ approach faces potential scalability problems since all routers must maintain per flow state.

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 application the appropriate resources are reserved and the level of performance of its connection is assured. The relative DiffServ model [8][9] provides QoS guarantees per service 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 provision define service differentiation qualitatively [10][11] 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 absolute service differentiation.

In absolute DiffServ, link sharing schedulers (e.g., Weighted Fair Queueing [12]) are used in the network infrastructure to provide guarantees of QoS values. Different bandwidth proportions are allocated to each service class, differentiating thus QoS provision. Most current implementations adjust service rates each time new requests are made, while the bandwidth portions are in general left static after admission control, regardless of the traffic dynamics [13]. However, taking into account the fact that most admission control schemes [14][15][16] consider average traffic arrival rate, in conjunction with the non static in general source's behaviour, congestion is likely to emerge particularly on the core network routers, constituting thus the aforementioned aspect inflexible, not scalable and often impractical. Some methods proposed (e.g., congestion control schemes) are adaptive in the sense of alleviating congestion in the network when detected, without however taking directly into consideration any feedback with respect to the QoS violation.

Considering the case of a high service class overloading (e.g., such may be the case of simultaneous activation of many applications of a high rated service class), worse packet forwarding may emerge with respect to the lower service classes. Additionally, at the same time no more applications/users could be accepted in those classes. This would be a case of *inconsistent* or *unpredictable* service class differentiation. The above scenarios belong to the case where one (or more) service classes utilise their allocated portion of bandwidth. Thus, an incremental differentiation to a source's traffic arrival rate or a request for a new application flow contributing to this (those) service class(es), exceeds the bandwidth reserved and thus, the extra generated traffic cannot be accommodated. In such a case, a dynamic service rate reconfiguration scheme of routers output link is required in order to provide the best QoS possible per flow, maximizing as well the utilization of the network. Consequently, assuming that 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, this paper proposes a consistent service differentiation provision on output links of core routers, so that the QoS levels required are satisfied.

Most methods proposed for dynamic bandwidth allocation utilize WFQ scheduler variances adjusting properly weights corresponding to the classes served taking into account traffic condition indicators, particular objectives and frequency of updates (e.g., packet arrival rate, queue length/buffer occupancy, delay measured, traffic patterns). Examples of such work include [17][18][19][20][21][22][23], while in [13] reinforcement learning is exploited in order to provide a solution to the bandwidth provisioning optimal control problem.

In the context of this paper, the problem of the improvement and fairness of *absolute* QoS provisioning in paths established along a DiffServ network is considered 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. In the current version of our study, service differentiation is defined in terms of local packet delays per service class and is provided through dynamic service rate reconfiguration on each output link of the router(s) along the respective path. This consideration is based on the fact that pi path's delay service class k packets are experiencing, which covers from an ingress to an egress node, is bounded by $D_{pi,k}$. In essence, this parameter 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.

Two versions of the problem are addressed. First, a confined version, which exploits the available resources of service classes in order to improve QoS characteristics of the problematic flows. Preliminary results of this method are presented in [24]. In the extended version of the problem, we consider an architecture, where each service class besides the absolute QoS parameters, is assigned with a relative importance factor, which in fact forms the basis of our extended service rate reconfiguration scheme. Specifically, in the extended version is considered of outmost importance to strictly meet the QoS parameters of the higher order service classes with respect to the lower ones. In order to accomplish this task the strict QoS constraints of the service classes rated less than the one(s) presenting the problem at a specific time instance during its(their) lifetime are relaxed and portion of their resources is allocated so that traffic of the higher rated problematic service classes is accommodated.

The paper is organised as follows. In section 2, a formal model of the service rate reconfiguration problem (both confined and extended version) is concisely defined. Section 3 presents a mathematical formulation of the service rate reconfiguration problem, while in Section 4, the problem is solved by means of a 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 rated 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 5, a set of results, indicative of the performance of our proposed reconfiguration scheme is presented. Finally, concluding remarks are made.

2 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 of 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 in reference with 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 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(es) 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. Two versions of the service rate reconfiguration problem will be formally defined and solved. The first one (*confined version*) assumes strict absolute QoS for all service classes. Thus, the delay related QoS constraint of the service classes not presenting a violation at time instance t should be preserved after applying the reconfiguration scheme, while the QoS characteristics of the rest service classes should be improved. The *extended service rate reconfiguration scheme* addressed in the context of this paper, assumes a hybrid (bearing resemblance to both Absolute and Relative) DiffServ architecture and relaxes the absolute QoS constraints of the lower-order service classes in order to satisfy the strict absolute QoS constraints of the higher order service classes.

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, $C = \{i \in SC \mid d(i,t) < d_g(i)\}$. 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, $j \in P$, respectively. 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 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 $p(i,j)$, $0 \leq p(i,j) \leq 1$ 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., a configuration of service rates $sr(i)$ to service classes $i \in SC$, which should maximise an objective 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 will be 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 changed to preservation of a minimum service rate. Thus, $sr(i) \neq 0,\ \forall i \in SC$.

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

Problem 1: [Service Rate Reconfiguration Problem-Confined Version]. Given:

(a) The set of service classes SC,

(b) The set of candidate service classes C for allocating portion of their resources to service class j, $(\forall j \in P)$,

(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,

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.

After introducing the extensions described above, the extended version of overall problem can be formally stated as follows.

Problem 2: [Service Rate Reconfiguration Problem - Extended Version]. Given:

(a) The set of service classes SC,

(b) The sets of candidate service classes C_j for allocating portion of their resources to service class j, $(\forall j \in P)$,

(c) the importance factor $RF(i)$ associated with each service class i,

(d) the cost factors $cf_u(i)$ and $cf_r(i)$ associated with each service class i,

(e) the service rate $sr_{pre}(i)$ attributed to service class i at time instance t,

(f) the absolute delay related QoS constraint $d_g(i)$ for each service class i,

(g) the packet delay $d_{pre}(i)$ encountered by service class i at time instance t,

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 $sr_{post}(i) \neq 0,\ \forall i \in SC$, and that all resources should be assigned to a service class.

The above general problem versions are open to various solution methods. Their 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.

3 OPTIMAL FORMULATION

In this sub-section the problems above are formulated as a 0-1 linear programming [25][26]. 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 $x(i, j, p_{ij})$ $(i \in C(C_j)$,

$j \in P$, $p_{ij} \in [0,1]$), 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.

Problem 1: [Service Rate Reconfiguration Problem - Confined Version].

Maximise:

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

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, $q(i)$, and the service rate allocated to service class i, $sr_{post}(i)$ and $sr_{pre}(i)$, respectively.

subject to

$$\sum_{j \in P \cup \{i\}} p_{ij} \cdot x(i, j, p_{ij}) = 1 \quad , \quad \forall i \in C \qquad (2)$$

$$d_{post}(i,t) \leq d_g(i) \quad , \quad \forall i \in C \qquad (3)$$

$$SRC(t) = \{sr_{post}(i) \mid i \in SC\} \qquad (4)$$

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

$$sr_{post}(i) = \begin{cases} sr_{pre}(i) \cdot [1 - \sum_{j \in P} p_{ij} \cdot x(i, j, p_{ij})], \ \psi(i) = 0 \\ sr_{pre}(i) + \sum_{j \in C} p_{ji} \cdot x(j, i, p_{ji}) \cdot sr_{pre}(j), \ \psi(i) = 1 \end{cases}, \ \forall i \in SC \quad (5)$$

$$b_c(SRC(t)) = \sum_{i \in SC} [d_{pre}(i) - d_{post}(i)] \cdot RF(i) \cdot \psi(i) \quad (6)$$

$$c_c(SRC(t)) = \sum_{i \in SC} [d_{post}(i) - d_{pre}(i)] \cdot cf_u(i) \cdot (1 - \psi(i)) \quad (7)$$

Relation (1) expresses the objective of finding the best assignment of service rates to service classes that maximises the objective 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 to set P, $b_c(SRC(t))$-relation (6), as well as the cost $c_c(SRC(t))$-relation (7), stemming from the deterioration of the quality of the service classes that assign portion of their resources to other service classes. Relations (4) and (5) express the new configuration scheme regarding service rates for each service class $i \ (i \in SC)$. Constraints (2) guarantee that each service class $i \ (i \in C)$ will assign all its resources. Constraints (3) guarantee 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.

For the extended version of the service rate reconfiguration problem, the decision variables $z(i)$, $\forall i \in SC$ are introduced, which assume the value 1(0) depending on whether for the service class i the delay related QoS absolute constraints are (are not) relaxed. The problem of obtaining the most appropriate configuration $SRC(t)$ according to the extended version may be obtained by reduction to the following optimisation problem.

Problem 2: [Service Rate Reconfiguration Problem - Extended Version].

Maximise:

$$\begin{aligned} f\left(t, SRC(t)\right) = & \sum_{i\in SC}[d_{pre}(i) - d_{post}(i)]\cdot RF(i)\cdot \psi(i) \\ & - \sum_{i\in SC}[d_{post}(i) - d_{pre}(i)]\cdot cf_u(i)\cdot(1-\psi(i))\cdot(1-z(i)) \\ & - \sum_{i\in SC}[d_{post}(i) - d_g(i)]\cdot cf_r(i)\cdot z(i) \qquad \forall i \in SC \end{aligned} \tag{8}$$

where, as in the confined version of the problem, $d_{post}(i)$, $d_{pre}(i)$ is the packet delay encountered by service class $i \in SC$ at time instance t, after/before applying the service rate reconfiguration scheme, respectively.

subject to
$$\sum_{j\in P\cup\{i\}} p_{ij}\cdot x(i,j,p_{ij}) = 1 \qquad \forall i \in C_j \tag{9}$$

$$sr_{post}(i) \neq 0 \qquad \forall i \in SC \tag{10}$$

$$SRC(t) = \{sr_{post}(i) \mid i \in SC\} \tag{11}$$

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

$$sr_{post}(i) = \begin{cases} sr_{pre}(i)\cdot[1-\sum_{j\in P} p_{ij}\cdot x(i,j,p_{ij})], & \psi(i)=0 \\ sr_{pre}(i) + \sum_{j\in C} p_{ji}\cdot x(j,i,p_{ji})\cdot sr_{pre}(j), & \psi(i)=1 \end{cases} \qquad \forall i \in SC \tag{12}$$

$$b_c(SRC(t)) = \sum_{i\in SC}[d_{pre}(i) - d_{post}(i)]\cdot RF(i)\cdot \psi(i) \tag{13}$$

$$\begin{aligned} c_c(SRC(t)) = & \sum_{i\in SC}[d_{post}(i) - d_{pre}(i)]\cdot cf_u(i)\cdot(1-\psi(i))\cdot(1-z(i)) \\ & + \sum_{i\in SC}[d_{post}(i) - d_g(i)]\cdot cf_r(i)\cdot z(i) \end{aligned} \tag{14}$$

Relation (8) expresses the objective of finding the best assignment of service rates to service classes that maximises the cost function, which is associated with the overall delay related QoS characteristics of the service classes. Relation (13) expresses the satisfaction stemming from the improvement of the quality of the service classes belonging in the set P, $b_c(SRC(t))$, while relation (14), provides the cost $c_c(SRC(t))$ illustrating in essence the deterioration of the quality of the service classes that assign portion of their resources to other service classes. As anticipated, the cost factor $cf_r(i)$ is contributing to the cost $c_c(SRC(t))$, in case the absolute QoS constraints of service class i are relaxed. Relations (11) and (12) express again the new configuration scheme regarding service rates for each service class i $(i \in SC)$. Constraints (9) guarantee that each service class i $(i \in SC)$ will assign all its resources, as was the case in the confined version of the problem. Constraints (10) guarantee that the service rate of the service class i $(i \in SC)$ will not ever obtain zero value.

4 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 [27][28], taboo search [29][30], genetic algorithms [31][32][33][34], 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 completed in an efficient and quick manner.

The algorithm adopted in this paper for the solution of the *service rate reconfiguration problem* (both the confined and the extended version) works as follows: As a first phase, the algorithm attempts to find a solution meeting the absolute QoS constraints for each service class. To this respect, the available resources of each service class are exploited in order to improve the QoS characteristics of the service classes presenting a QoS constraint violation. In case a solution at this phase (which in essence forms the *confined version* of the problem) is not feasible, the algorithm moves at a second phase, where the strict constraints of the lower rated service classes are relaxed in order to gain the absolute constraints of the higher rated service classes (which in essence covers the *extended version* of the service rate reconfiguration problem). The aforementioned approach is illustrated graphically in Figure 1. At this point it should be noted that the core of the service rate reallocation procedure for the confined and the extended version of the service rate reconfiguration problem (*Step 4* and *Step 6* of Figure 1, respectively) follows the *simulated annealing technique* and its description is provided in the following subsection.

4.1 ALGORITHM BASED ON SIMULATED ANNEALING

Annealing is the physical process in which a crystal is cooled from the liquid to the solid state. Careful cooling brings the crystal in the minimum energy state. In analogy, a simulated annealing algorithm considers each solution of the optimisation problem as a state, the cost of each solution as the energy of the state, and the optimal solution as the minimum energy state.

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 $x(i, j, p_{ij})$, 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.

The simulated annealing-based algorithm may be described as follows.

Basic Simulated Annealing Algorithm.

Step 0. Initialisation. Get an initial solution, IS, and an initial temperature value T. The currently best solution (CBS) is IS, i.e., $CBS = IS$, and the current temperature value (CT) is T, i.e., $CT = T$.

Step 1. If $CT = 0$, or if the stop criterion is satisfied, the procedure ends and a transition to *step 6* is performed.

Step 2. A new solution (NS) that is neighbouring to CBS is found.

Step 3. The difference of the costs of the two solutions, CBS and NS is found, i.e., the quantity $\Delta c = C(CBS) - C(NS)$ is computed.

Step 4. If $\Delta c \le 0$ then the new solution becomes the currently best solution, i.e., $CBS = NS$. Otherwise, if $\Delta c > 0$, then if $e^{-(\Delta c / CT)} > rand[0,1)$, the new solution becomes the currently best solution, i.e., $CBS = NS$.

Step 5. The cooling schedule is applied, in order to calculate the new current temperature value CT, and a transition to *step 1* is performed.

Step 6. End.

There are various alternatives for realising the stop criterion mentioned in step 1. In our version, the algorithm stops when no improvement in the objective function has been achieved after a given number of temperature decreases (in other words consecutive moves or alterations of the currently best solution). Neighbouring solutions (step 2) are selected randomly among all the neighbouring ones of the currently best solution, with the same probability for all neighbours.

5 RESULTS

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 two versions of the service rate reconfiguration problem that should be solved in the context of the IP network QoS management.

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 given aim at the provision of indicative evidence of the efficiency and the effectiveness of the proposed service rate reconfiguration schemes (both the confined and the extended version). Following two sets of experiments will be used for demonstrating these aspects. The first experiment entails the simplest possible case, as it comprises only one router. This consideration will be changed in the second set of experiments. Nevertheless, in the first experiment it enables the acquisition of an initial set of indicative results that show the behaviour of our schemes. In order to test the performance of the proposed framework of this paper we used NS2 network simulator [35] 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.

The schemes introduced in this paper may be realised by means of new service components realising the aforementioned logic easily integrated with the existing network elements. Mobile Intelligent Agent Technology [36] may be adopted for the implementation of the *Service Rate Reconfiguration Module* (*SRRM*) forming part of a

Configuration and Reconfiguration Management Subsystem, which support the network administrator in a real network environment. Mobile intelligent agents performing the reconfiguration tasks should be sent from the management station as close as possible to the routers so as to complete their mission. *SRRM* caters for the initial configuration of the routers (i.e., definition of the parameters for the DiffServ service classes on the basis of past experience/ historical data). Additionally, it monitors the network load and QoS conditions and adjusts the service rates in case the predefined thresholds are about to be exceeded, while in parallel a notification is sent to the network operator. Thus, action is taken before potential SLAs are breached and the customer's experience is adversely affected. The aforementioned implementation is considered as a standalone issue, left for future study.

Figure 2 shows the topology adopted for the realisation of the first 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 source nodes with shape parameter $\alpha = 1.9$ and mean on and off time 5msec. This choice was driven by the fact that, even in aggregate, Pareto sources exhibit highly burstly characteristics. The packet length of incoming traffic is taken equal to 1Kbyte for all classes.

Packets are passed from their sources to their destinations through a CBQ scheduler [37] with output link capacity of 10Mbps. CBQ is a link sharing scheduler which is variation of the GPS algorithm [38]. 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 the most appealing advanced scheduler available today used to provide differentiated services.

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 reconfiguration scheme achieves the required absolute delay differentiation through the dynamic adaptation of service rates per service class. Specifically, the service rates are dynamically adjusted when the predefined threshold values d_g are violated. In our approach we considered the application of the reconfiguration scheme after regular time intervals denoted as U. A small U would increase processing load in the routers, and thus, the computational complexity of the system. A large U would result in packet delay approximations per service class that will not conform to the real ones. As in [18], we find that for $0.001\text{sec} \leq U \leq 0.1\text{sec}$ the behavior of the scheme is good. In our experiments, U is taken equal to 0.1sec. This time period is considered by the authors to be quite small, covering, thus, cases of dynamic traffic sources variations, without increasing the computational complexity of the routers. At this point it should be noted that our proposed scheme based on the simulated annealing paradigm provides good (near optimal) solutions within the time limits required (less than time period U), so as to adequately handle extreme variations to the traffic load originated by the sources and satisfy the absolute constraints of the service classes.

Table 1 presents the values of all input parameters (i.e., Relative Importance Factor, cost of allocating part of a service class's available resources, service rate allocated to each service class prior to the application of the reconfiguration scheme, and delay guaranteed per service class) considered in the experiment. In the reconfiguration experiments performed the local delay experienced by each service class i packets prior as well as after the application of the reconfiguration scheme, $d_{pre}(i)$ and $d_{post}(i)$, respectively, is estimated on the basis of the equation (15) within the regular time intervals U. 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:

$$d_{post,pre}(i) = \frac{q(i)}{sr_{post,pre}(i)} \qquad , \qquad \forall i \in SC \qquad (15)$$

Figure 3 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.

Both the static and dynamic cases have been considered in the experiment conducted. Figure 4(a-b) presents in a graphical manner the local delays experienced by the packets of each service class after the application of the reconfiguration scheme, concerning the static and the dynamic case, respectively. Regarding the static case, as it can be observed, the required absolute delay differentiation is satisfied for the individual packet delays per service class as long as the percentage of input traffic per class of service does not change. The service rate values $sr_{post}(i)$ are adjusted once on the basis of equation (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.

In the dynamic case, the reconfiguration scheme achieves the required absolute delay differentiation through the dynamic adaptation of service rates per service class. Thus, all service classes i achieve average and individual packet delays, which are below the predefined threshold value $d_g(i)$.

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%. Figure 5(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. From the obtained results of Figure 4 and Figure 5, we could say that if the average packet delay experienced by service class i is 25msec less than $d_g(i)$ then individual packet delays satisfy the related delay constraints.

Following, in the context of the first set of experiments, the required delay of the fourth class is changed from 80 to 60msec ($d_g(4) = 60$ msec). As a first step, the dynamic service rate reconfiguration scheme is applied in order to achieve the required absolute delay differentiation with $sr_{post}(i)$ values computed using (1). Figure 6(a) presents in a graphical manner the local delays experienced by the packets of each service class after the service rate reconfiguration has been carried out. Figure 6(b) depicts the required delay constraints, the average packet delay prior and after the reconfiguration scheme has been performed. From the obtained results it may be noted that the average packet delay parameter presents an overall improvement of approximately 40%.

However as illustrated in Figure 6(a), the absolute delay constraint values can not be satisfied for individual packets of *service classes 3* and *4*. Specifically, most packets of *class 3* experience delays which are above 150msec and some packets of *class 4* experience delays which exceed the required constraint of 60 msec, even though their average delay is small enough compared to $d_g(4)$. For the packets of *class 3* this can be also concluded from Figure 6(b) in which the average packet delay for packets of class 3 when reconfiguration is applied converges to $d_g(3)$. Since absolute delay constraints can not be satisfied for all classes of service, the extended version of the algorithm should be exploited so that strict constraints of lower classes will be relaxed in favor of the higher rated service classes. Table 2 lists the cost factor values of each service class for providing not available resources and their required minimum service rate utilized in the extended algorithm. The rest input parameters of the algorithm remain unchanged with respect to Table 1.

Dynamic reconfiguration is performed while the $sr_{post}(i)$ values are calculated so that the objective function (9) is maximized. The absolute delay constraints $d_g(i)$ of the lower rated service classes are sacrificed in favor of the higher rated *service classes 3* and *4*. The results from this simulation (Figure 7) reveal that the required absolute delay differentiation is satisfied for classes 3 and 4. The rest two service classes acquire their minimum service rate as defined in Table 2 in favor of the other two higher rated classes. Considering the average packet delay each service class is experiencing prior to and after the application of the extended version of the reconfiguration scheme, it may be noted that this parameter presents an overall improvement of approximately 57%.

The topology adopted for the second set of experiments is illustrated in Figure 8. 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, through *routers 1* and *2,* which are assumed to be CBQ schedulers, as was also the case in the first set of experiments. The source nodes follow a Pareto distribution with shape parameter $\alpha = 1.9$ giving total input load 97% to *router 1*. Four Cross-Traffic sources on *router 2* send *class 1*, *class 2*, *class 3* and *class 4* classified packets of *Constant Bit Rate* (*CBR*) traffic each to the related destination sink, with rate adjusted so that the input load on this router reaches 97% utilization as well. The same assumptions and input parameters for each node have been considered as with the first set of experiments (Table 1). Average and individual end-to-end packet delays per service class originated from *router 1* will be measured. The absolute end-to-end delay constraint for each service class is equal to the sum of the respective $d_g(i)$ of each node.

The dynamic service rate reconfiguration scheme is applied on the output link of both routers. Figure 9 presents the local delays experienced by packets of each service class prior to and after the reconfiguration has taken place. Similarly, Figure 10 depicts the delay related QoS constraint and the average packet delay experienced by each service class prior to and after the reconfiguration across routers has been performed.

Figures 9 and 10 show that the required absolute end to end delay differentiation is achieved for all service classes after the application of the reconfiguration scheme. This is satisfied for the average packet delay as well as for the individual packet delays of successive packets. 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 24%.

At this point it should be noted that end to end differentiation through static reconfiguration can not be performed instantaneously on all nodes across a network path. This is owed to the fact that after the first router reconfiguration has taken place significant changes may exist to the input loads of the following routers across the path.

In such a case packet delay approximations per class of service in the next nodes can not be estimated using the current queue loads of classes through formulae (15). Thus, the administrator after reconfiguring the first node should wait for a specified time period until the input loads of the rest nodes along the path are stabilized. Thereafter, a static reconfiguration request should be issued to the next node and so forth until all the nodes across the path are reconfigured.

6 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 of a router’s output link that should be solved in the context of IP network management framework. Two versions of the problem are addressed, the confined and the extended. The confined version of the reconfiguration scheme exploits the available resources of service classes in order to satisfy the absolute QoS constraints of a service class that its own resources prove to be inadequate for the satisfaction of its absolute delay constraints. In the extended version, the accomplishment of strict QoS constraints for the higher order classes is achieved by relaxing the strict QoS constraints of the service classes rated less than the one(s) presenting the problem, covering the cases where the confined version of the algorithm prove to be inadequate. Finally, end to end delay measurements have been performed across a path of routers providing indicative evidence of the service rate reconfiguration scheme effectiveness and efficiency.

Directions for future work include, but are not limited to the following. First, the experimentation with various network load conditions, particularly in cases where input loads to service classes vary. Second, the realisation of further wide scale trials in a real network environment, so as to experiment with the applicability of the framework presented herewith.

7 REFERENCES

[1] P. P White, “RSVP and Integrated Services in the Internet: A Tutorial”, *IEEE Communications Magazine*, pp. 100-106, May 1997.

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

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

[8] C. Dovrolis, P. Ramanathan, “A Case for Relative Differentiated Services and the Proportional Differentiation Model”, *IEEE Network*, Vol. 13 No. 5, pp. 26-34, Sept. 1999.

[9] C. Dovrolis, D. Stiliadis, “Relative Differentiated Services in the Internet: Issues and Mechanisms”, *ACM SIGMETRICS Performance Evaluation Review* Vol. 27, No 1, pp. 204-205, June 1999.

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

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

[12] A. Demars, S. Keshav, S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm”, In Proc. of ACM SIGCOMM’89, Sept. 1989.

[13] T.C.K. Hui and C.K. Tham, “Adaptive Provisioning of Differentiated Services Networks based on Reinforcement Learning”, *IEEE Transactions on Systems, Man & Cybernetics*, Vol. 33, No. 4, pp. 492-501, Nov. 2003.

[14] B. Choi, R. Bettati, “Efficient Resource Management for Hard Real-Time Communication over Differentiated Services Architecture”, In Proc. of the 7th IEEE RTCSA, Dec. 2000.

[15] C. Cetinkaya, V. Kanodia, and E. Knightly “Scalable Services via Egress Admission Control”, *IEEE Transactions on multimedia*: Special Issue on Multimedia over IP, Vol. 3, No. 1, March 2001.

[16] L. Breslau, E. W. Knightly, S. Shenker, I. Stoica, and H. Zhang, “Endpoint admission control: Architectural issues and performance”, In Proc. of ACM SIGCOMM'00, pages 57-69, Sept. 2000.

[17] C. C. Li, S.-L. Tsao, M. C. Chen, Y. Sun, and Y.-M. Huang, “Proportional Delay Differentiation Service Based on Weighted Fair Queueing”, In Proc. IEEE International Conference on Computer Communications and Networks (ICCCN), Oct. 2000.

[18] M-F. Horng, et al, “An Adaptive Approach to Weighted Fair Queue with QoS Enhanced on IP Network”, In Proc. IEEE TENCON 2001, Aug. 2001.

[19] D. Hang, et al, “TD^{2}FQ: An Integrated Traffic Scheduling and Shaping Scheme for DiffServ Networks”, In Proc. IEEE HPSR 2001, May 2001

[20] H. Wang, C. Chen, K.G. Shin, "Adaptive-Weighted Packet Scheduling for Premium Service", In Proc. IEEE ICC 2001, June 2001.

[21] R-F. Liao, A. T. Campbell, "Dynamic Core Provisioning for Quantitative Differentiated Service", In Proc. IEEE IWQoS 2001, June 2001

[22] A. Michalas, V. Fafali, M. Louta, G. Karetsos, V. Loumos, "Proportional Delay Differentiation Provision by Bandwidth Adaptation of Class Based Queue Scheduling", *International Journal of Communication Systems*, Vol. 17, Issue 7, pp. 743-761, July 2004.

[23] A. Michalas, P. Fafali, M. Louta, V. Loumos, "Proportional Delay Differentiation Employing the CBQ Service Discipline", In Proc. 7th International Conference on Telecommunications (ConTel 2003), June 2003.

[24] M. Louta, A. Michalas, E. Loutas, V. Loumos, "Quality of Service Management in IP networks", In Proc. 7th International Conference on Telecommunications (ConTel 2003), June 2003.

[25] H. Salkin, "Integer programming", Addison-Wesley, Reading, Massachusetts, 1975.

[26] C. Papadimitriou, K. Steiglitz, "Combinatorial optimization: Algorithms and complexity", Prentice Hall, Inc., 1982.

[27] E. Aarts, J. Korts, "Simulated annealing and the Boltzmann machines", Wiley, New York, 1989.

[28] P. van Laarhoven, E. Aarts, "Simulated annealing: Theory and applications", Kluwer Academic Publisher, Dordrecht, 1988.

[29] F. Glover, M. Laguna, "Taboo search", Kluwer Academic Publisher, Hingham, Massachusetts, 1998.

[30] F. Glover, E. Taillard, D. de Werra, "A User's Guide to Taboo Search", Annals of Operations Research, Vol. 41, 1993.

[31] D. Goldbgerg, “Genetic algorithms in search, optimisation and machine learning”, Addison-Wesley, Reading, Massachusetts, 1989.

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

[33] R. Haupt, S. Haupt, “Practical genetic algorithms”, J. Wiley & Sons, New York, 1997.

[34] Z. Michalewicz, “Generic algorithms + Data structures = Evolution programs”, Springer-Verlag, Berlin, 1995.

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

[36] A. Bieszczad, T. White, and B. Pagurek, “Mobile Agents for Network Management”, In IEEE Communications Surveys, Fourth Quarter 1998, Vol. 1, No. 1, Sept. 1998.

[37] S. Floyd and V. Jacobson, “Link-sharing and resource management models for packet networks”, *IEEE/ACM Transactions on Networkin*g, Vol. 3 No. 4, pp. 365–386, Aug. 1995.

[38] H. Zhang, “Service disciplines for guaranteed performance service in packet-switching networks”, In IEEE Proceedings, Vol. 83, pp. 1374-1399, Oct. 1995.

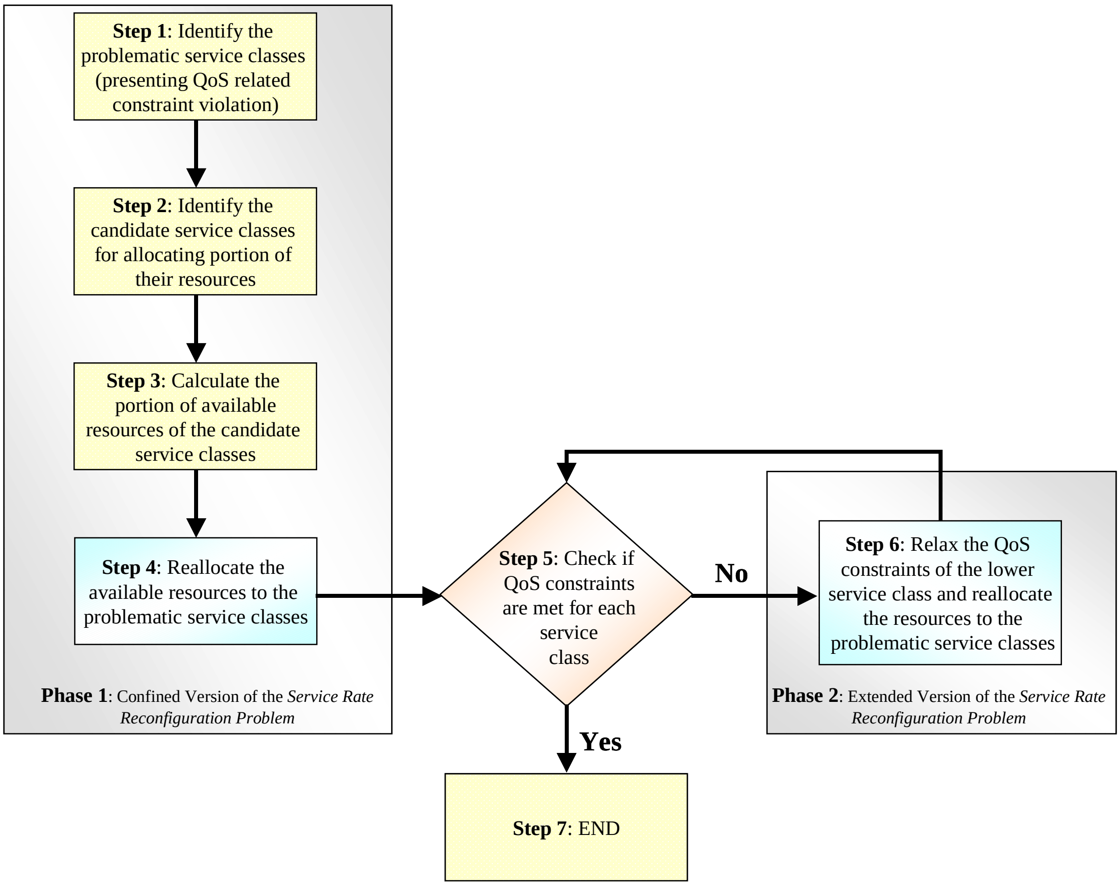


Figure 1: Description of the algorithm adopted for the solution of the *Service Rate Reconfiguration problem*.

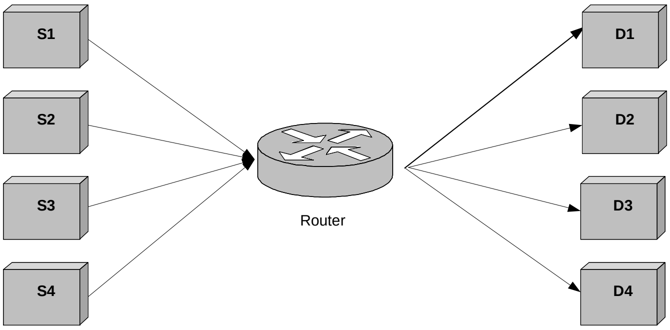


Figure 2. Simulation topology for the first set of experiments

Parameters \ *classes*	*class1*	*class2*	*class3*	*class4*
Generated Traffic(Mb)	1.708	3.826	5.933	7.935
$RF(i)$	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

Table 1. Input parameters for the first experiment

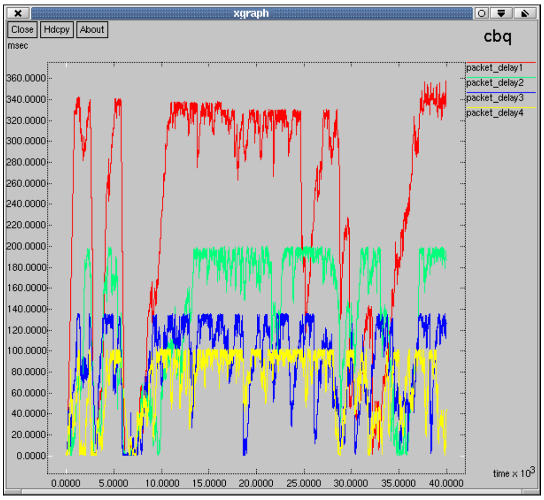
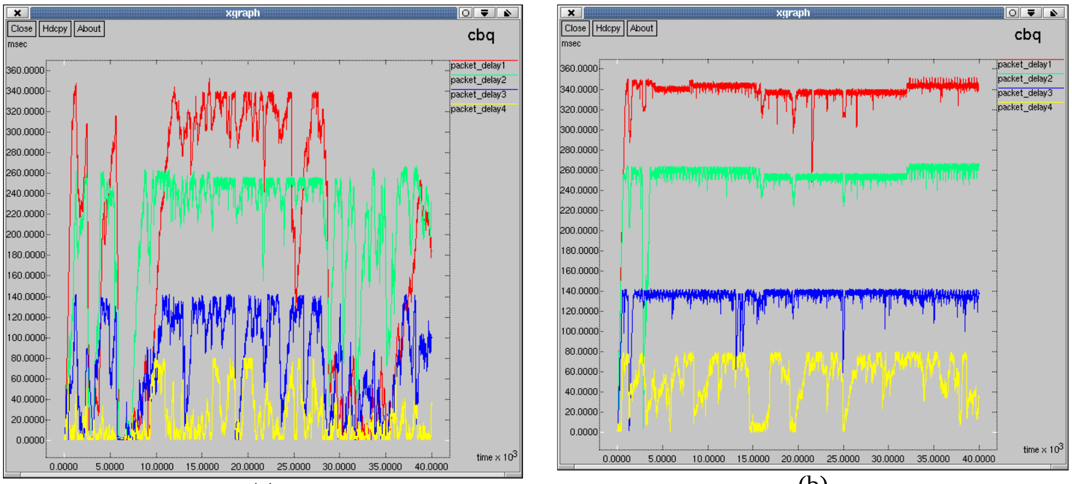
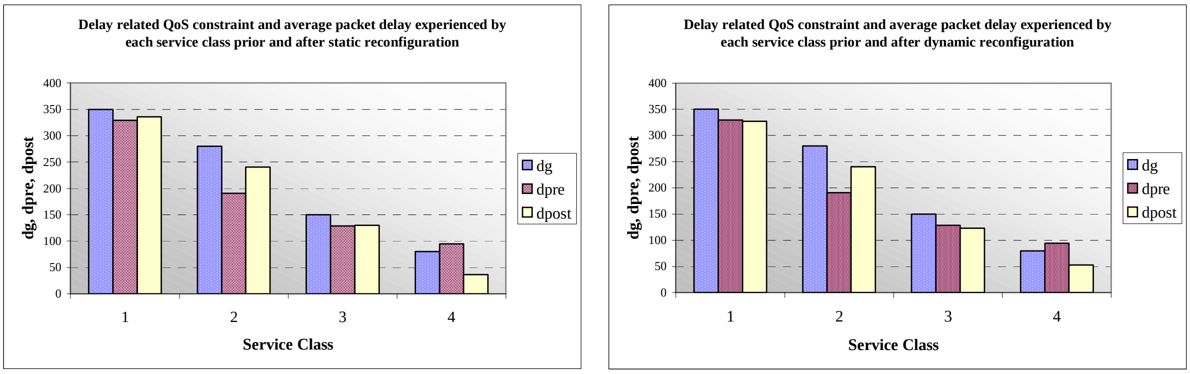


Figure 3. Individual delays of successive packets prior to reconfiguration



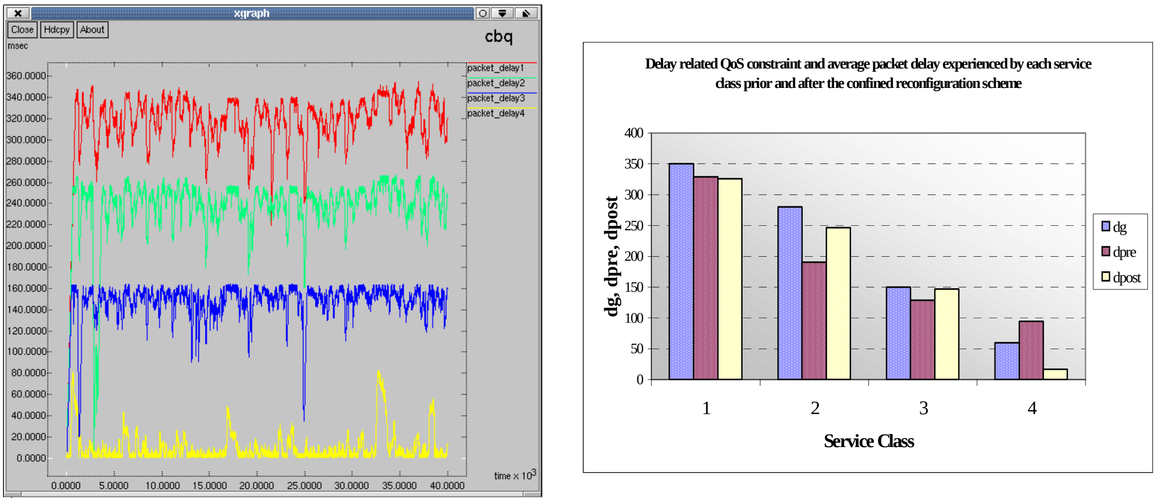
(a) (b)

Figure 4(a-b). Individual delays of successive packets utilizing the reconfiguration scheme considering (a) the static and (b) the dynamic case, respectively.



(a) (b)

Figure 5(a-b). 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.

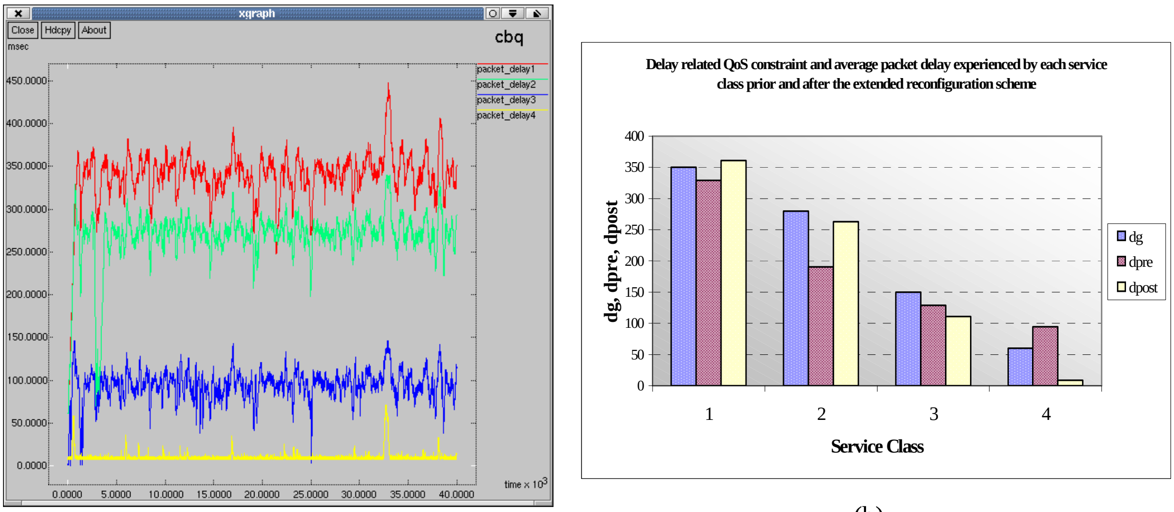


(a) (b)

Figure 6(a-b). Dynamic service rate reconfiguration using the confined version of the algorithm. (a) Individual delays of successive packets after the application of the reconfiguration scheme and (b) Delay related QoS constraint and average packet delay prior and after the reconfiguration scheme has been applied.

Parameters \ *classes*	*class1*	*class2*	*class3*	*class4*
Generated Traffic(Mb)	1.708	3.826	5.933	7.935
$RF(i)$	1	10	100	1000
$cf_u(i)$	0.0001	0.001	0.01	0.1
$cf_r(i)$	0.001	0.01	0.1	1
$sr_{pre}(i)$ *(Mb)*	1	2	3	4
$d_g(i)$ *(msec)*	350	280	150	60
Minimum Service Rate *(Mb)*	0.5	1.3	1.7	2

Table 2. Input parameters used in the extended service rate reconfiguration scheme.



(a) (b)

Figure 7. Dynamic service rate reconfiguration using the extended version of the algorithm. (a)Individual delays of successive packets while reconfiguration is performed and (b) Delay related QoS constraint and average packet delay prior and after the reconfiguration has been applied.

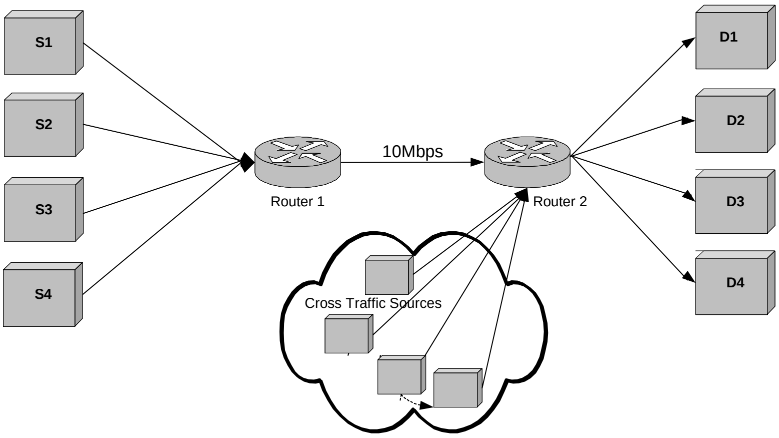
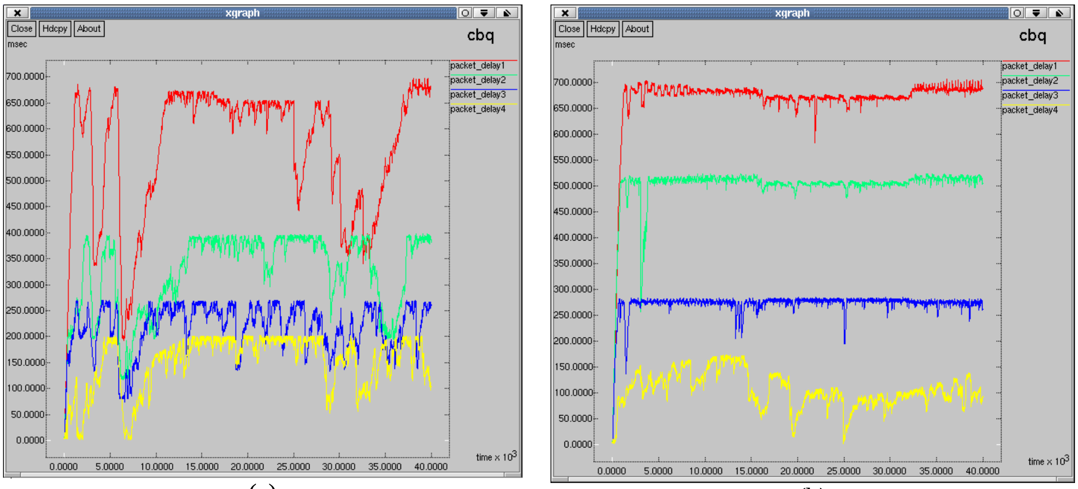


Figure 8. Simulation topology for the second set of experiments. Traffic is traversing two CBQ nodes.



(a) (b)

Figure 9(a-b). Individual end to end delays of successive packets (a) prior to the reconfiguration scheme and (b) after reconfiguration has been applied.

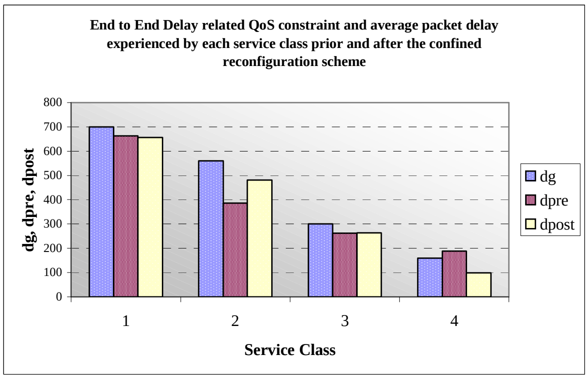


Figure 10. End to End delay related QoS constraint and average packet delay prior to and after the reconfiguration on both routers across the path has been applied.