A New Preemption Policy for DiffServ-Aware Traffic Engineering to Minimize Rerouting

J. C. de Oliveira, C. Scoglio, I. F. Akyildiz, and G. Uhl

Abstract—**In this paper, a new preemption policy is proposed and complemented with an adaptive scheme that aims to minimize rerouting. The preemption policy combines the three main optimization criteria: number of LSPs to be preempted, priority of LSPs to be preempted, and amount of bandwidth to be preempted. The preemption policy is complemented by an adaptive scheme that selects LSPs with lower priority and reduces their rate in order to accommodate the new high-priority LSP setup request. Heuristics for both preemption and adaptive preemption policies are derived. Simulation results show the heuristics' accuracy. Performance comparisons of a non-preemptive approach, our preemption policy, the adaptive rate policy, and the policy in use by commercial routers are included.**

I. Introduction

ONE of the most actively studied open issues in several areas of communication networks is the problem of bandwidth reservation and management. The objective is to maximize the network resources utilization while minimizing the number of connections that would be denied access due to insufficient resource availability. Load balancing is another important issue. It is desirable to avoid that portions of the network become overutilized and congested, while alternate feasible paths remain underutilized. These issues are addressed by Traffic Engineering [1].

Multiprotocol Label Switching (MPLS) is a switching technology to forward packets based on a short, fixed length *label*. Using indexing instead of long address matching, MPLS performs fast forwarding. MPLS creates Label Switched Paths (LSPs); packets with identical label are forwarded on the same LSP [2]. An extension of the Resource Reservation Protocol (RSVP) is used to establish and maintain LSPs in the backbone [3]. One of the most significant applications of MPLS will be in Traffic Engineering (TE), since LSPs can be considered as virtual traffic trunks carrying flow aggregates generated by packet classification [4].

In IETF RFC 2702, [1], issues and requirements for Traffic Engineering in an MPLS network are highlighted. In order to address both traffic oriented and resource oriented performance objectives, the authors point out the need for *priority* and *preemption* parameters as Traffic Engineering attributes of traffic trunks. A *traffic trunk* is an aggregate of traffic flows belonging to the same class. Traffic trunks are routable objects, distinct from the LSP which they traverse, and are unidirectional. A request for resources for a new traffic trunk implies the setup of a new LSP.

The *preemption attribute* determines whether an LSP with a certain *priority attribute* can preempt another LSP with a lower *priority attribute* from a given path, when there is a competition for available resources. The preempted LSP may then be rerouted. Preemption can be used to assure that high priority LSPs can be always routed through relatively favorable paths within a differentiated services environment. In the same context, preemption can be used to implement various prioritized access policies as well as restoration policies following fault events [1]. Preemption policies have also been recently proposed in other contexts. In [5], the authors developed a framework to implement preemption policies in non-Markovian Stochastic Petri Nets (SPNs). In a computing system context, preemption has been applied in cache-related events. In [6] a technique to bound cache-related preemption delay is proposed. Finally, in the wireless mobile networks framework, preemption has been applied to handoff schemes [7].

Although not a mandatory attribute in the traditional IP world, preemption becomes indeed a more attractive strategy in a differentiated services scenario [8], [9]. Moreover, in the emerging optical network architectures, preemption policies can be used to reduce restoration time for high priority traffic trunks under fault conditions. Nevertheless, in the DiffServ-aware Traffic Engineering (DS-TE) approach, whose issues and requirements are discussed in [10], the preemption policy is again considered an important piece on the bandwidth reservation and management puzzle, but no preemption strategy is defined.

In this paper, a new preemption policy is proposed and complemented with an adaptive scheme that aims to minimize rerouting. The preemption policy is both simple and robust, combining the three main optimization criteria: number of LSPs to be preempted, priority of LSPs to be preempted, and amount of bandwidth to be preempted. Using our policy, a service provider can balance the objective function that will be optimized in order to stress the desired criteria. The preemption policy is complemented by an adaptive scheme that selects lower priority LSPs that can afford to reduce their rate. The selected LSPs will fairly reduce their rate in order to accommodate the new high-priority LSP setup request. Heuristics for both simple preemption policy and adaptive preemption scheme are derived. Simulation results show the heuristics' accuracy. Performance comparisons among a non-preemptive approach, our new preemption policy, and our new policy combined with the adaptive scheme are also provided.

The rest of this paper is organized as follows: In Section II, we introduce the preemption problem. A mathematical formulation, a simple heuristic, and simulation results for the proposed preemption problem are discussed in Section III. In Section IV, we extend the proposed preemption policy by complementing it with the adaptive rate scheme. A mathematical formulation for the optimization problem, a simple heuristic, and example results are included in this section. Performance evaluation of the proposed policy, the policy complemented by the adaptive scheme, and a non-preemptive approach are discussed in Section V. Finally, the paper is concluded in Section VI.

J. C. de Oliveira, C. Scoglio, and I. F. Akyildiz are with the Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. E-mail: {jau, caterina, ian}@ece.gatech.edu.

G. Uhl is with Swales Aerospace Technology and Scientific Services, Lanham, MD. E-mail: uhl@rattler.gsfc.nasa.gov.

II. Problem Formulation

In this section we present the preemption problem formulation in a per-class Traffic Engineering (TE) context.

Existing TE mechanisms only support constraint based routing of traffic based on a single bandwidth constraint [1]. No per-class treatment is allowed. In DiffServ-aware Traffic Engineering (DS-TE), traffic flows can be grouped into classes, and different bandwidth constraints can be applied to each class [10]. By mapping a traffic trunk in a given class on a separate LSP, DS-TE allows the traffic trunk to utilize resources available to that class on both shortest and non-shortest paths, and follow paths that meet the specific constrains of that given class.

Performing Traffic Engineering on a per-class basis requires that certain parameters are propagated via Internet Gateway Protocol (IGP) link state advertisements (LSAs). In order to reduce the number of parameters to be advertised, information is propagated on a *per-Class-Type* instead of on a per-class basis. A *Class-Type* is composed by classes with similar aggregate maximum and minimum bandwidth requirements. There is no enforced minimum or maximum bandwidth requirement at the individual class level. As a consequence, IGP needs to advertise separate Traffic Engineering information for each Class-Type, which consists of the *Unreserved Bandwidth* (UB) information [10]. This information will be used as a per Class-Type bandwidth constraint for Constraint Based Routing and also for admission control purposes.

For each Class-Type, there exist 8 preemption levels ranging from 0 to 7. The lower number represents higher priority, i.e., an LSP with preemption level 0 can preempt all other LSPs with non-zero preemption level value. Eight classes are currently supported by DS-TE and may be grouped into Class-Types [11]. In order to provide different bandwidth constraints for each Class-Type, a configurable maximum reservable bandwidth for each or for an aggregate of Class-Types must be defined. Lower priority Class-Types (e.g. Best Effort) should not be completely starved by higher priority classes. Therefore, where N Class-Types CT_i $(i = 0, \cdots, N-1)$, ordered with respect with their priorities (lowest CT_i has lowest preemption level. CT_0 is the lowest priority, therefore assigned to Best Effort traffic), are supported, the following bandwidth constraints may be configured on a given link by the network operator [10]:

- No more than $P_{N-1}\%$ of CT_{N-1};
- No more than $P_{N-2}\%$ of $\mathrm{CT}_{N-1} + \mathrm{CT}_{N-2}$;
- $\cdots$
- No more than $P_0\%$ of $\mathrm{CT}_{N-1} + \mathrm{CT}_{N-2} + \cdots + \mathrm{CT}_0$;

where $P_{N-1}, P_{N-2}, \cdots, P_0$ are percentages of the total link capacity, each separately configurable for every link, and $P_{N-1} \leq P_{N-2} \leq \cdots \leq P_0$, $P_0 = 100\%$.

Define **UB** as a vector with dimension N, with each component UB(i), representing the Unreserved Bandwidth for CT_i. When a new LSP setup request arrives for an active Class-Type CT_i, in a network for which N Class-Types are active, the **UB** components UB(i) are updated as follows:

$$\mathrm{UB}(i) = P_i\%\, C - \sum_{k=i}^{N} b(\mathrm{CT}_k),$$

where C is the capacity of the link shared by the Class-Types and $b(\mathrm{CT}_k)$ represents the bandwidth in use by Class-Type CT_k.

For example, suppose a link with 1500 capacity units and only 2 active CTs: CT_1, voice traffic with priority 2; and CT_0, data traffic with priority 4. Suppose P_1=50% and P_0=100%. The UB initial advertisement can be illustrated in a vector as $[750, 1500]$. The advertisement shows the Unreserved Bandwidth for each active Class-Type, at the initial state, with no traffic passing through the link.

Suppose the link is in the initial state and 4 consecutive requests arrive: 500 of CT_0, 500 of CT_1, 250 of CT_0, and 250 of CT_1. The UB advertisements are as shown:
$[750, 1500]$ initially;
$[750, 1000]$ after 500 of CT_0 arrive;
$[250, 500]$ after 500 of CT_1 arrive;
$[250, 250]$ after 250 of CT_0 arrive;
$[0, 0]$ after 250 of CT_1 arrive.
Note that no preemption is needed in this case.

Instead, if 3 requests for LSP setup arrive in sequence: 750 of CT_0, another 750 of CT_0, and 500 of CT_1, the UB advertisements are:
$[750, 1500]$ initially;
$[750, 750]$ after 750 of CT_0 arrive;
$[750, 0]$ after another 750 of CT_0 arrive;
$[250, 0]$ after 500 of CT_1 arrive.
Note that the last step requires that 500 capacity units are preempted from CT_0.

It is important to mention that preemption can be reduced if an alternative shortest-path route (e.g., second or third shortest-path) can be considered. Even in that case, preemption may be needed as such path options may also be congested. In a fixed shortest-path routing approach, preemption would happen more frequently.

In the case in which preemption will occur, a preemption policy should be activated to find the preemtable LSPs with lower preemption levels.

Now an interesting question arises: which LSPs should be preempted? Running preemption experiments using commercial routers, we could conclude that the preempted LSPs were always the ones with the lowest priority, even when the bandwidth allocated was much larger than the one required for the new LSP. This policy would result in high bandwidth wastage for cases in which rerouting is not allowed. An LSP with a large bandwidth share would be preempted to give room to a higher priority LSP that requires a much lower bandwidth.

A new LSP setup request has two important parameters: bandwidth and preemption level. In order to minimize wastage, the set of LSPs to be preempted can be selected by optimizing an objective function that represents these two parameters, and the number of LSPs to be preempted. More specifically, the objective function could be any or a combination of the following [8], [9], [12]:

1. Preempt the connections that have the least priority (preemption level). The QoS of high priority traffics would be better satisfied.
2. Preempt the least number of LSPs. The number of LSPs that need to be rerouted would be lower.
3. Preempt the least amount of bandwidth that still satisfies the request. Resource utilization would be better.

After the preemption selection phase is finished, the selected LSPs must be torn down (and possibly rerouted), releasing the reserved bandwidth. The new LSP is established, using the currently available bandwidth. The UB information is then updated. Fig. 1 shows a flowchart that summarizes how each LSP setup request is treated in a preemption enabled scenario.

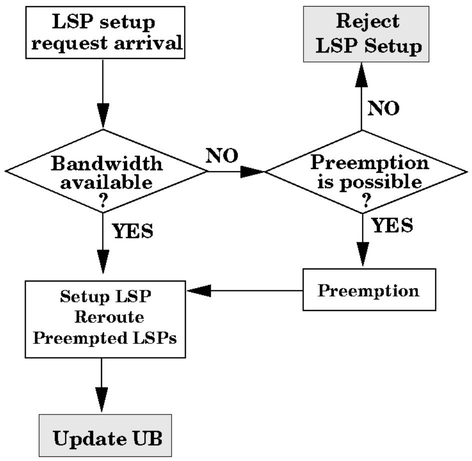

Fig. 1. Flowchart for LSP setup procedure.

In [12], the authors propose connection preemption policies that optimize the discussed criteria in a given order of importance: number of connections, bandwidth, and priority; and bandwidth, priority, and number of connections. The novelty in our approach is to propose an objective function that can be adjusted by the service provider in order to stress the desired criteria. No particular criteria order is enforced. Moreover, our preemption policy is complemented by an adaptive rate scheme. The resulting policy reduces the number of preempted LSPs by adjusting the rate of selected low-priority LSPs in order to accommodate a higher-priority request. This approach minimizes service disruption and rerouting decision and signaling.

III. Preemption Policy

In this section, a new mathematical formulation for the preemption problem is presented. A simple heuristic is proposed, and simulations results are shown to compare both approaches.

A. Preempting Resources on a Path

It is important to note that once a request for an LSP setup arrives, the routers on the path to be taken by the new LSP need to check for bandwidth availability in all links that compose the path. For the links in which the available bandwidth is not enough, the preemption policy needs to be activated in order to guarantee the end-to-end bandwidth reservation for the new LSP. This is a decentralized approach, in which every node on the path would be responsible to run the preemption algorithm and determine which LSPs would be preempted in order to fit the new request. A decentralized approach may sometimes not lead to an optimal solution.

Another idea would be to use a centralized approach, in which a "manager entity" would run the preemption policy and determine the best LSPs to be preempted in order to free the required bandwidth in all the links that compose the path. A unique LSP may be already set in between several nodes on that path, and the preemption of that LSP would free the required bandwidth in many links that compose the path.

For this paper, we choose to use the decentralized approach, which is easier to be integrated to the current Internet protocol. The parameters required by the policy are currently available for protocols such as OSPF or are easy to be determined.

B. Mathematical Formulation

We formulate the preemption policy problem with an integer optimization approach. Consider a request for a new LSP setup with bandwidth b and a certain preemption level. When preemption is needed, due to lack of available resources, the preemptable LSPs will be chosen among the ones with lower preemption level in order to fit $b - \mathrm{UB}_0$ bandwidth units, where UB_0 represents the Unreserved Bandwidth with respect to the lowest priority traffic, which corresponds to the unused bandwidth on that link. We define a constant r such that $r = b - \mathrm{UB}_0$, representing the actual bandwidth that needs to be preempted (the required minus the available bandwidth).

Without loss of generality, we assume that bandwidth is available in bandwidth modules, which implies that variables such as r and b are integers.

Define $\mathcal{L}$ as the set of active LSPs having a priority level lower than the LSP setup request. We denote the cardinality of $\mathcal{L}$ by L. $b(l)$ is the bandwidth reserved by LSP $l \in \mathcal{L}$, expressed in bandwidth modules and $p(l)$ is the priority level of LSP l.

In order to represent a cost for each preemption level, we define an associated cost $y(l)$ inversely related to the preemption level $p(l)$. For example, we can choose a linear relation $y(l) = 8 - p(l)$. We define $\mathbf{y}$ as a cost vector with L components, $y(l)$. We also define $\mathbf{b}$ as a reserved bandwidth vector with dimension L, and components $b(l)$.

The vector $\mathbf{z}$ is the optimization variable. $\mathbf{z}$ is composed by L binary variables, each defined as follows:

$$z(l) = \begin{cases} 1 & \text{if LSP } l \text{ is preempted;} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

For example, assume there exists three LSPs, $\mathcal{L} = (l_1, l_2, l_3)$, with reserved bandwidth of 3 Mbps, 2 Mbps, and 1 Mbps, respectively. Consider that $p(l_1) = 3$, $p(l_2) = 6$, $p(l_3) = 7$. Consequently $y(l_1) = 5$, $y(l_2) = 2$, $y(l_3) = 1$, $\mathbf{y} = [5, 2, 1]$, and $\mathbf{b} = [3, 2, 1]$. $\mathbf{z} = [0, 1, 1]$ means that LSPs l_2 and l_3 are preempted.

Concerning the objective function, as reported in the previous section, three main objectives can be reached in the selection of preempted LSPs:

- minimize the priority of preempted LSPs,
- minimize the number of preempted LSPs,
- minimize the preempted bandwidth.

To have the widest choice on the overall objective that each service provider needs to achieve, we define the following objective function F, which for simplicity is chosen as a weighted sum of the above mentioned criteria:

$$F(\mathbf{z}) = \alpha(\mathbf{z} \cdot \mathbf{y}^{\mathrm{T}}) + \beta(\mathbf{z} \cdot \mathbf{1}^{\mathrm{T}}) + \gamma(\mathbf{z} \cdot \mathbf{b}^{\mathrm{T}}) \quad (2)$$

where the term $\mathbf{z} \cdot \mathbf{y}^{\mathrm{T}}$ represents the preemption level or priority of preempted LSPs, $\mathbf{z} \cdot \mathbf{1}^{\mathrm{T}}$ represents the number of preempted LSPs ($\mathbf{1}$ is a unit vector with adequate dimension), and $\mathbf{z} \cdot \mathbf{b}^{\mathrm{T}}$ represents the total preempted capacity. Coefficients α, β, and γ

are suitable weights that can be configured in order to stress the importance of each component in F.

The following constraint ensures that the preempted LSPs release enough bandwidth to satisfy the new request:

$$\mathbf{z} \cdot \mathbf{b}^{\mathbf{T}} \geq \mathbf{r} \quad (3)$$

Fig. 2 contains a summary of the proposed integer program.

GIVEN	$\mathcal{L}, \mathbf{b}, \mathbf{y}, r, \alpha, \beta, \gamma$.
FIND	$\mathbf{z}$ (L binary integer variables)
MINIMIZING	$F(\mathbf{z}) = \alpha(\mathbf{z} \cdot \mathbf{y}^{\mathbf{T}}) + \beta(\mathbf{z} \cdot \mathbf{1}^{\mathbf{T}}) + \gamma(\mathbf{z} \cdot \mathbf{b}^{\mathbf{T}})$
SUBJECT TO	$\mathbf{z} \cdot \mathbf{b}^{\mathbf{T}} \geq \mathbf{r}$

Fig. 2. Our preemption policy formulation.

C. Preemption Heuristic

The choice of LSPs to be preempted is known to be an NP-complete problem [13]. For networks of small and medium size, or for a small number of LSPs, the online use of an optimization tool is a fast and accurate way to find the solution. However, for large networks and large number of LSPs, a simple heuristic that could approximate the optimal result would be preferable.

In order to simplify the online choice of LSPs to be preempted, we propose the following equation, used in our heuristic (Fig. 3):

$$H(l) = \alpha y(l) + \beta + \gamma(b(l) - r)^2 \quad (4)$$

In this equation, $\alpha y(l)$ represents the cost of preempting LSP l, β represents the choice of a minimum number of LSPs to be preempted in order to fit the request r, and $\gamma(b(l)-r)^2$ penalizes a choice of an LSP to be preempted that would result in high bandwidth wastage.

In our heuristic, H is calculated for each LSP. The LSPs to be preempted are chosen as the ones with smaller H that add enough bandwidth to accommodate r. The respective components in the vector $\mathbf{z}$ are made equal to one for the selected LSPs.

In case H contained repeated values, the sequence of choice follows the bandwidth b reserved for each of the regarded LSPs, in increasing order. For each LSP with repeated H, we test whether the bandwidth b assigned to that LSP only is enough to satisfy r. If there is no such LSP, we test whether the bandwidth of each of those LSPs, added to the previously preempted LSPs' bandwidth is enough to satisfy r. If that is not true for any traffic trunk in that repeated H value sequence, we preempt the LSP that has the larger amount of bandwidth in the sequence, and keep preempting in decreasing order of b until r is satisfied or the sequence is finished. If the sequence is finished and r is not satisfied, we again select LSPs to be preempted based on an increasing order of H. More details on the algorithm to implement our heuristic, called **Preemption**, are shown in Fig. 3.

After finishing the algorithm, $\mathbf{z}$ contains the information about which LSPs are to be preempted and *preempt* contains the amount of bandwidth preempted. In a much larger network, our heuristic would still be very simple to compute when compared to the optimization problem described in equations 2 and 3.

```
Algorithm Preemption (L, b, y, r, α, β, γ)
preempt = 0;
if Σ_∀i b(i) >= r,
  calculate H(i) = αy(i) + β + γ(b(i) − r)^2;
  sort H, b, in increasing order of H(i);
  if H has repeated values sort those by increasing b;
  while preempt < r
    if b(i) < r
      if H does not have repeated values
        preempt in increasing order of H.
        preempt = preempt+b(i); z(i) = 1;
      else
        if b(i) >= r for any repeated H
          preempt = b(i); % preempt only i
          z(i) = 1; % make all other z equal to zero; break;
        elseif preempt+b(i) > r for any repeated H
          preempt = preempt + b(i);
          z(i) = 1; break;
        else
          preempt larger b(i) in the sequence and add b(i − 1)
          until preempt >= r or repeated sequence is finished.
          preempt = preempt + b(i);
          z(i) = 1; i = i − 1;
        end
      end
    else
      preempt=b(i);
      z(i) = 1; % make all other z equal to zero; break;
    end
  end
else
  reject LSP setup request
end
return(z)
```

Fig. 3. Heuristic for our preemption policy.

D. Example

Consider a network composed by 16 LSPs with reserved bandwidth $\mathbf{b}$ in Mbps and cost $\mathbf{y}$, as shown in Table I.

TABLE I
BANDWIDTH AND COST INFORMATION FOR SIMULATIONS.

Traffic trunk	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8
Bandwidth ($\mathbf{b}$)	20	10	60	25	20	1	75	45
Cost ($\mathbf{y}$)	7	6	5	4	3	2	1	3
Traffic trunk	l_9	l_{10}	l_{11}	l_{12}	l_{13}	l_{14}	l_{15}	l_{16}
Bandwidth ($\mathbf{b}$)	100	5	40	85	50	20	70	25
Cost ($\mathbf{y}$)	5	2	4	3	6	5	4	1

Suppose the network operator decides to configure $\beta = 0$, indicating that the number of trunks preempted is not important (rerouting is allowed and not expensive: small topology), $\alpha = 1$ and $\gamma = 1$, indicating that preemption level and preempted bandwidth are more important.

A request for an LSP establishment arrives with $r = 155$ Mbps and $p = 0$ (highest possible priority). From equations 2 and 3, we formulate the following optimization problem:

Minimize $F(\mathbf{z}) = (\mathbf{z} \cdot \mathbf{y}^{\mathbf{T}}) + (\mathbf{z} \cdot \mathbf{b}^{\mathbf{T}})$,

subject to $\mathbf{z} \cdot \mathbf{b}^{\mathbf{T}} \geq \mathbf{155}$, with $\mathbf{y}$ and $\mathbf{b}$ defined as in Table I.

Using an optimization tool such as LINDO or CPLEX to solve the above optimization problem, one will find that traffic trunks l_8, l_{12}, and l_{16} are selected for preemption.

Suppose the network operator decides that it is more appropriate to configure $\alpha = 1$, $\beta = 1$, and $\gamma = 0$, because in this network rerouting is now cheaper, LSP priority is again very important, but bandwidth is not a critical issue. The optimization problem now becomes:

Minimize $F(\mathbf{z}) = (\mathbf{z} \cdot \mathbf{y^T}) + (\mathbf{z} \cdot \mathbf{1^T})$,

subject to $\mathbf{z} \cdot \mathbf{b^T} \geq \mathbf{155}$, in which case, LSPs l_7 and l_{12} are selected for preemption.

To take into account the number of LSPs preempted, the preemption level, and the amount of bandwidth preempted, the network operator may set $\alpha = \beta = \gamma = 1$. In that case, LSPs l_{12} and l_{15} are selected.

From the above example we can observe that when the number of traffic trunks preempted was not an issue, 3 LSPs adding exactly the requested bandwidth, and with the lowest priority were selected. When a possible waste of bandwidth is not an issue, 2 LSPs were selected, adding more bandwidth than requested, but with lower priority. Considering the three factors as crucial, 2 LSPs are preempted, and in this case adding exactly 155 Mbps with the lowest possible priorities.

Using the same data as in Table I, and with $\alpha = \beta = \gamma = 1$, we varied the value of the request r and compared the results found by our optimization formulation (Fig. 2) and our heuristic (Fig. 3), regarding the final cost achieved, calculated by equation 2. Fig. 4 shows the result of these tests.

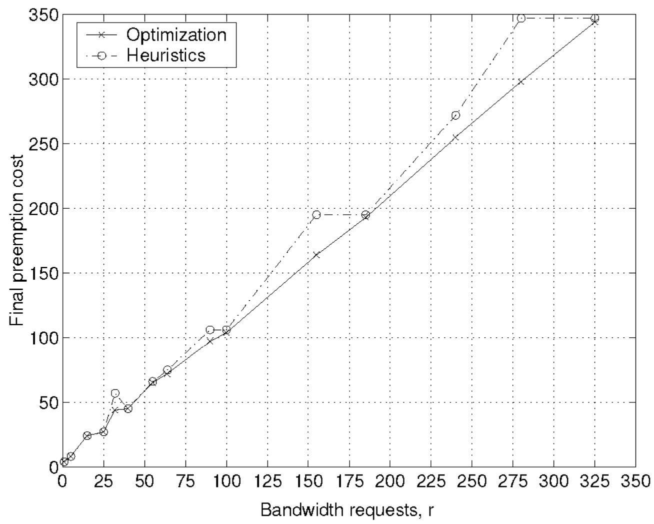

Fig. 4. Comparison between optimization formulation and heuristic.

Figures 5, 6, and 7, show results for the heuristic and optimization problem when only the priority, only the number of LSPs preempted, or only the amount of preempted bandwidth is important, respectively.

Figures 5 and 6 show the perfect accuracy of the heuristic for the considered cases. The results in Fig. 4 and in Fig. 7 show that the heuristic finds a similar solution for most of the cases, and that when r increases the heuristic leads to a slightly higher cost - a price paid because the heuristic follows a much simpler approach. When comparing the bandwidth request r to the already setup bandwidth reservations, we observe that when r is comparable to one or two LSP reservations (which is a more likely scenario) the heuristic always result in a similar cost solution. The zig-zag effect on the graphic is due to the preemption

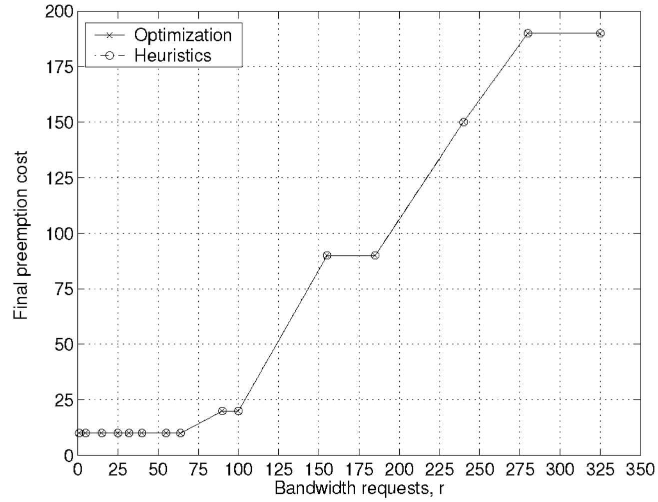

Fig. 5. Optimization formulation and heuristic when $\alpha = 10$, $\beta = 0$, $\gamma = 0$.

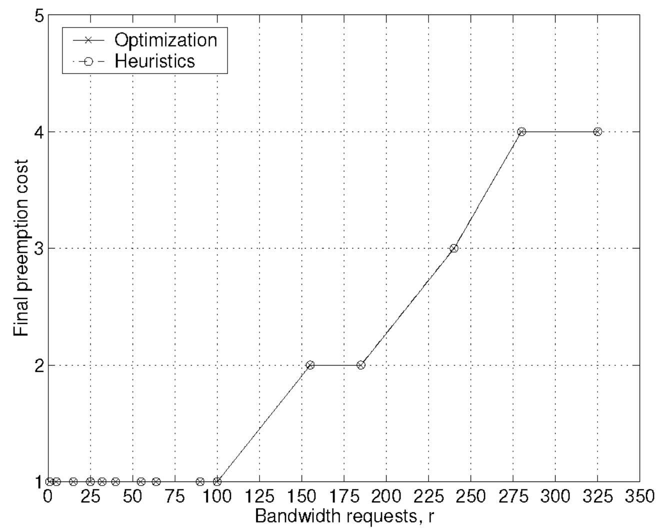

Fig. 6. Optimization formulation and heuristic when $\alpha = 0$, $\beta = 1$, $\gamma = 0$.

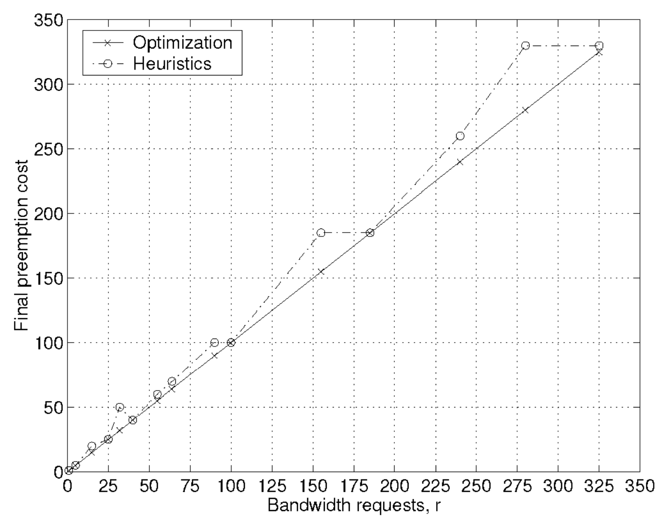

Fig. 7. Optimization formulation and heuristic when $\alpha = 0$, $\beta = 0$, $\gamma = 1$.

of LSPs that add more bandwidth than the request r, which increases the value found by the cost function. When the next request r is considered, and the new selected traffic trunks add exactly or about the same value as the new r, the cost function is reduced, therefore the zig-zag occurs.

IV. Preemption Policy with Adaptive Rate Scheme

In this section we complement our new preemption policy with an adaptive rate scheme. In the previous section, when a set of LSPs was chosen to be preempted, those LSPs were torn down and could be rerouted, which implied extra signaling and routing decisions. In order to avoid or minimize rerouting, we propose to reduce the number of preempted LSPs by selecting a few low-priority traffic trunks that would have their rate reduced by a certain maximum percentage in order to accommodate the new request. In the future, whenever there exists available bandwidth in the network, the lowered-rate traffic trunks would fairly increase their rate to the original reserved bandwidth.

Some applications such as non-real-time video or data transfer can afford to have their transmission rate reduced, and would be the most likely to be assigned to such Class-Types and preemption levels. By reducing the rate in a fair fashion, the LSPs would not be torn down, there would not be service disruption, extra setup and torn down signaling, or rerouting decisions. For the DiffServ technology, traffic aggregates assigned to the Assured Forward Per-Hop Behavior (AF PHB) would be the natural candidates for rate reduction. Whereas Expedited Forward Per-Hop Behavior (EF PHB) supports services with "hard" bandwidth and jitter guarantees, the AF PHB allows for more flexible and dynamic sharing of network resources, supporting the "soft" bandwidth and loss guarantees appropriated for bursty traffic [14].

Next, we present a mathematical formulation for our scheme, followed by a simple heuristic that approximates the results provided by the optimization problem. Simulation results are shown to stress the accuracy of the proposed heuristic. Comparison with the preemption policy simulation results of the previous section are also included, taking into account costs and rewards of both approaches.

A. Rate Reduction on a Path

We chose to use a decentralized approach to solve the rate reduction problem on a path (link by link). Every node on the path would be responsible to run the rate reduction algorithm and determine which LSPs would have their rate reduced in order to fit the new request. Similarly to the preemption problem, a centralized approach could be implemented in this case.

B. Mathematical Formulation

Similarly to the preemption policy, again we formulate the preemption policy as an integer optimization problem.

We assume that bandwidth is available in bandwidth modules, and define $\mathcal{L}'$ (cardinality L') as a set of active traffic trunks with priority lower than the LSP setup request, and that can afford to have their rate reduced. Therefore, $\mathcal{L}' \subset \mathcal{L}$. The parameters r, $b(l)$, and $p(l)$ have the same context as in Section III.

We define M as the total number of bandwidth modules allocated to LSPs that can be preempted or have their rate reduced:

$$M = \sum_{l=1}^{L'} b(l) \tag{5}$$

We also define the vector $\mathbf{v^l}$ with M components, representing the bandwidth modules reserved by an active LSP $l \in \mathcal{L}'$: $\mathbf{v^l} = (v_1^l, v_2^l, \cdots, v_M^l)$, where

$$v_m^l = \begin{cases} 1 & \text{if module } m \text{ belongs to } l; \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

$\mathbf{V}$ is a matrix, $\mathcal{L}' \times M$, composed by vectors $\mathbf{v^l}$. We define the vector $\mathbf{B}$ as $\mathbf{B} = \mathbf{V} \cdot \mathbf{b}$

We again define $\mathbf{y}$ as a priority vector, now with M components, where each component $y(m)$ is the priority y of the bandwidth module m. Every bandwidth module of an LSP has the same cost value, which implies that $\mathbf{y}$ is composed by a serie of repeated values (as many as the number of modules in each LSP). Vectors $\mathbf{x}$ and $\mathbf{z}$ are the variables to be optimized, and are defined as follows.

Vector $\mathbf{x}$ is composed by M binary variables:

$$x(m) = \begin{cases} 1 & \text{if module } m \text{ is preempted;} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

A binary component $x(m) = 1$ means that the m-th bandwidth module is preempted in order to reduce that LSP's rate and make room to satisfy the request of r bandwidth modules.

Vector $\mathbf{z}$ is composed by L' binary variables, and follows the same definition as in Section III (equation 1).

Note that the optimization variables are binary and their total number is $L' + M$.

For example, assume there exists three LSPs that can afford to have their rate reduced, $\mathcal{L}' = (l_1, l_2, l_3)$, with reserved bandwidth of 3 Mbps, 2 Mbps, and 1 Mbps, respectively. Assume that the bandwidth module is 1 Mbps. The size of the set of bandwidth modules can be calculated with equation 5: $M = 6$. Each LSP can be represented by the following bandwidth module vectors (equation 6):
$\mathbf{v^{l_1}} = (1,1,1,0,0,0) \rightarrow$ modules 1, 2, and 3 belong to l_1.
$\mathbf{v^{l_2}} = (0,0,0,1,1,0) \rightarrow$ modules 4 and 5 belong to l_2.
$\mathbf{v^{l_3}} = (0,0,0,0,0,1) \rightarrow$ module 6 belongs to l_3. Lets assume that $p(l_1) = 3$, $p(l_2) = 6$, $p(l_3) = 7$. Consequently $y(l_1) = 5$, $y(l_2) = 2$, $y(l_3) = 1$ and $\mathbf{y} = (5,5,5,2,2,1)$.

We define the following new objective function $\mathcal{F}$:

$$\mathcal{F}(\mathbf{x},\mathbf{z}) = \alpha(\mathbf{x} \cdot \mathbf{y^T}) + \beta(\mathbf{z} \cdot \mathbf{1^T}) + \gamma(\mathbf{x} \cdot \mathbf{1^T}) + \mathbf{x} \cdot (\mathbf{1}/\mathbf{B})^\mathbf{T} \tag{8}$$

where $\mathbf{x} \cdot \mathbf{y^T}$ represents the priority of preempted bandwidth modules, $\mathbf{z} \cdot \mathbf{1^T}$ represents the number of preempted LSPs, $\mathbf{x} \cdot \mathbf{1^T}$ represents the total preempted capacity, and $\mathbf{x} \cdot (\mathbf{1}/\mathbf{b})^\mathbf{T}$ represents the bandwidth module cost per LSP, proportional to the number of modules reserved by the LSP. Coefficients α, β and γ are used for the same purpose as in Section III (equation 2): in order to stress the importance of each component in $\mathcal{F}$.

As for constraints, we must make sure that the bandwidth requirement is met, that all the bandwidth modules from an LSP are made available when that LSP is preempted, that the respective modules for the LSPs that will reduce their rate are also preempted, and that the preempted rate will not be more than $\Delta\%$ of the reserved bandwidth for that LSP.

We represent these constraints as follows, remarking that the *greater than* and *less than* signs are considered in a row by row relation between the matrices:

$$
\begin{array}{lcll}
\mathbf{x}\cdot\mathbf{1^T} & \geq & \mathbf{r} & \textit{(1 constraint)} \\
\mathbf{x^T}-(\mathbf{z}\cdot\mathbf{V})^\mathbf{T} & \geq & 0 & \textit{(M constraints)} \\
\mathbf{z^T}-\mathbf{V}\cdot\mathbf{x^T} & > & -(b^T-1) & \textit{(L' constraints)} \\
\mathbf{V}\cdot\mathbf{x^T} & \leq & \frac{\Delta}{100}(\mathit{diag}(\mathbf{b^T}\cdot\mathbf{z})+\mathbf{b^T}) & \textit{(L' constraints)}
\end{array}
\quad (9)
$$

where *diag* means the diagonal values of the matrix displayed in a column vector.

The first constraint implies that the number of preempted capacity modules should be equal or greater than the request. The remaining constraints imply that when an LSP is preempted, all the capacity modules belonging to it should be preempted, that the respective modules for the LSPs that will reduce their rate are also preempted, and that the preempted rate is no more than $\Delta\%$ of the actual reserved bandwidth for that LSP. The total number of constraints is $2L' + M + 1$.

Fig. 8 contains a summary of the proposed integer program.

GIVEN	$\mathcal{L}', \mathbf{b}, \mathbf{y}, r, \alpha, \beta, \gamma, \Delta.$
FIND	$\mathbf{x}, \mathbf{z}$
MINIMIZING	$\mathcal{F}(\mathbf{x},\mathbf{z}) = \alpha(\mathbf{x}\cdot\mathbf{y^T}) + \beta(\mathbf{z}\cdot\mathbf{1^T}) + \gamma(\mathbf{x}\cdot\mathbf{1^T}) + \mathbf{x}\cdot(1/\mathbf{B})^\mathbf{T}$
SUBJECT TO	$\mathbf{x}\cdot\mathbf{1^T} \geq \mathbf{r}$
	$\mathbf{x^T}-(\mathbf{z}\cdot\mathbf{V})^\mathbf{T} \geq \mathbf{0}$
	$\mathbf{z^T}-\mathbf{V}\cdot\mathbf{x^T} > -(\mathbf{b^T}-\mathbf{1})$
	$\mathbf{V}\cdot\mathbf{x^T} \leq \frac{\Delta}{100}(\mathit{diag}(\mathbf{b^T}\cdot\mathbf{z})+\mathbf{b^T})$

Fig. 8. Adaptive preemption policy formulation.

C. *Prept_Reduce Heuristic*

As discussed before, the choice of LSPs to be preempted or to have their rate reduced is an NP-complete problem. In order to simplify and expedite the online choice of LSPs for preemption, we propose to use a simple heuristic shown to be as accurate as the optimization formulation illustrated in Fig. 8.

When using the adaptive scheme, a new LSP setup request is treated differently. First, we test whether there is enough bandwidth among the preemptable LSPs in order to fit the new request r. If $\sum_{i=1}^{\mathcal{L}} b(i) \geq r$, we proceed. If that is not true, the LSP setup request is rejected. Suppose there is enough bandwidth. Now we test whether there are LSPs that can afford to reduce their rate. If not, we run our **Preemption** algorithm illustrated in Fig. 3 and choose the LSPs that will be preempted and rerouted. If $\mathcal{L}' \neq \emptyset$, we test whether the bandwidth occupied by these LSPs is enough to fit r. If yes, we run the adaptive policy **Prept_Reduce** (Fig. 10), which will be explained in detail in the following, and choose the LSPs that will reduce their rate by a maximum of $\Delta\%$ or that will be completely preempted in order to accommodate r. If the bandwidth allocated to the LSPs that can reduce their rate is not enough to fit r, we execute the algorithm **Preemption** to choose one LSP to be preempted and test again if the remaining required bandwidth, $r-\mathit{preempt}$ can be made free by reducing the rate of the LSPs in $\mathcal{L}'$. Yet, if there is not enough resources, we run again **Preemption**, this time preempting two LSPs. Another test is made to see whether the remaining bandwidth can be accommodated by reducing the rate of $\mathcal{L}'$ elements and so on. Fig. 9 illustrates the new LSP setup procedure.

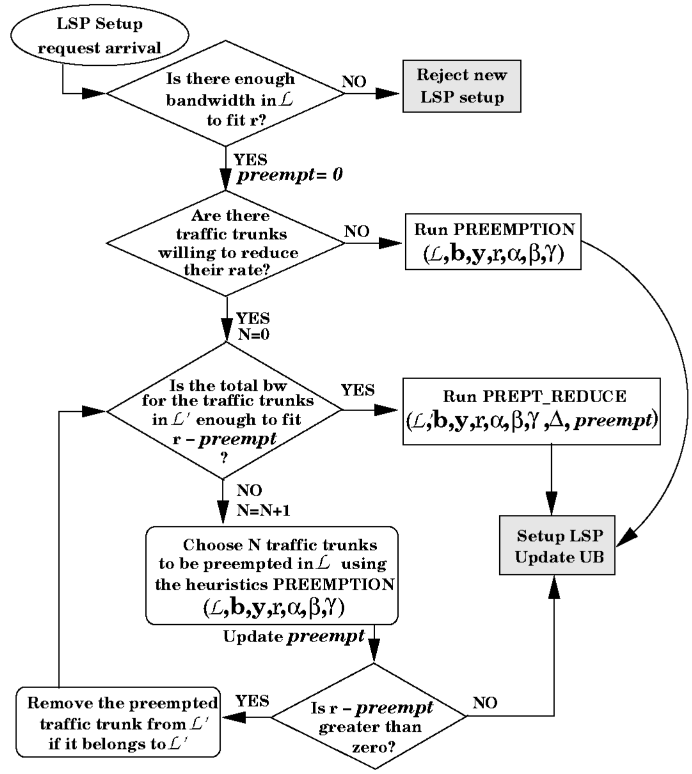

Fig. 9. Flowchart for LSP setup procedure with adaptive preemption policy.

We propose the following equation, used in our algorithm **Prept_Reduce** (Fig. 10):

$$\mathcal{H}(l) = \alpha y(l) + \beta + \gamma + 1/b(l) \quad (10)$$

In this equation, $\alpha y(l)$ represents the cost of preempting an LSP, β represents the choice of a minimum number of LSPs for preemption or rate-reduction, γ represents the amount of bandwidth to be preempted, and $1/b(l)$ represents an additional cost by bandwidth module. This cost is calculated as the inverse of the amount of bandwidth reserved for the considered LSP. In this way, an LSP with more bandwidth modules will be more likely to be preempted than one with just a few number of modules.

Our heuristic for the adaptive rate scheme uses $\mathcal{H}$ to determine the choice of bandwidth modules that will be preempted (sometimes resulting in a whole LSP being preempted). $\mathcal{H}$ is calculated for each LSP in $\mathcal{L}'$ and it does not depend on the request value r. Again, $\mathcal{H}$ is sorted in increasing order with repeated values being ordered by increasing associated bandwidth b.

Fig. 10 illustrates the heuristic algorithm. The following input data is given: the set $\mathcal{L}'$; the request r; the parameters α, β, γ; the vectors $\mathbf{b}$ and $\mathbf{y}$ containing bandwidth and cost information for each LSP in $\mathcal{L}'$, respectively; the maximum reduced rate per LSP in percentage Δ; and the amount of bandwidth already preempted in a previous ran **Preemption** algorithm, *preempt*. The algorithm calculates $\mathcal{H}$ and the preemptable bandwidth (number of modules) for each LSP, $\mathit{limit}(l) = \text{round\_down}(\Delta/100 * b(l))$: no more than $\Delta\%$ of $b(l)$.

Following the sorted $\mathbf{b}$ vector, and while the total preempted amount is not larger or equal to r, we test whether $b(i)$ is less than $r-$ *preempt*. If that is true, the whole LSP i will be preempted, $z(i)$ will be set to 1, and all the respective x modules will be also made equal to 1. If not, we preempt module by module of LSP i until we reach either the amount requested or the maximum value: *limit*(i). The vector $\mathbf{x}$ is always updated with the respective preempted modules being set to 1. After that, if the requested amount is still not reached, we choose a new LSP from the sorted $\mathcal{H}$ and repeat the described process.

After finishing the algorithm, $\mathbf{z}$ and $\mathbf{x}$ contain information about which LSPs are to be preempted or have their rate reduced and by how much. **Prept_Reduce** is very accurate, as shown in the example discussed next.

Algorithm Prept_Reduce ($\mathcal{L}', \mathbf{b}, \mathbf{y}, r, \alpha, \beta, \gamma, \Delta,$ *preempt*)
rate_reduce = 0;
generate vectors $\mathbf{x}$ and $\mathbf{x}$_index;
% $\mathbf{x}$_index contains pointers to where each LSP
% bandwidth allocation starts in vector $\mathbf{x}$.
limit = round_down($\Delta/100 * \mathbf{b}'$); % maximum rate reduction
calculate $\mathcal{H}(i) = \alpha y(i) + \beta + \gamma + 1/b(i)$;
sort $\mathcal{H}$, $\mathbf{b}$, in increasing order of $\mathcal{H}(i)$;
if $\mathcal{H}$ has repeated values sort those by increasing $\mathbf{b}$;
r_aux = r-*preempt*;
while *rate_reduce* + *preempt* $< r$
  if $b(i) <= r$_aux
    preempt = *preempt* + $b(i)$;
    r_aux = r_aux - $b(i)$; $z(i) = 1$;
  else
    index = $\mathbf{x}$_index(i);
    while bandwidth reduced for $i <$ *limit*(i) and
    rate_reduce + *preempt* $< r$
      x(index) = 1;
      rate_reduce = *rate_reduce* + 1;
      index = index + 1;
      r_aux = r_aux - *rate_reduce*;
    end
  end
end
return($\mathbf{x}, \mathbf{z}$)

Fig. 10. Heuristic for our preemption policy with adaptive scheme.

D. Example

Consider the same network proposed in Section III-D. Now suppose that LSPs l_1, l_2, l_6, l_{10}, and l_{13} are not available for rate reducing, which means that $\mathcal{L}' = [l_3, l_4, l_5, l_7, l_8, l_9, l_{11}, l_{12}, l_{14}, l_{15}, l_{16}]$. The vectors $\mathbf{b}$ and $\mathbf{y}$ are now composed by the bandwidth and cost assignments to these LSPs only.

We run several simulations varying the value of r in order to compare the cost of rate reduction and preemption, and with $\alpha = \beta = \gamma = 1$, which means that the network operator considered all the criteria with the same importance. Moreover, the parameter Δ was configured as $\Delta = 50$, indicating that each LSP in $\mathcal{L}'$ is willing to have its rate reduced by a maximum 50% of the bandwidth already reserved for it. Fig. 11 shows a chart that illustrates the results obtained for several different requests r. The x labels indicate that the rate of that LSP was reduced, while the z labels indicate that the whole LSP was preempted to satisfy the request r. Note that rate reduction never overcomes the 50% total bandwidth limit on each LSP.

Fig. 12, shows the accuracy of our adaptive-rate heuristic. The heuristic obtains the same results found by the optimization formulation. Several other cases were ran and the results found by the optimization and heuristic always matched.

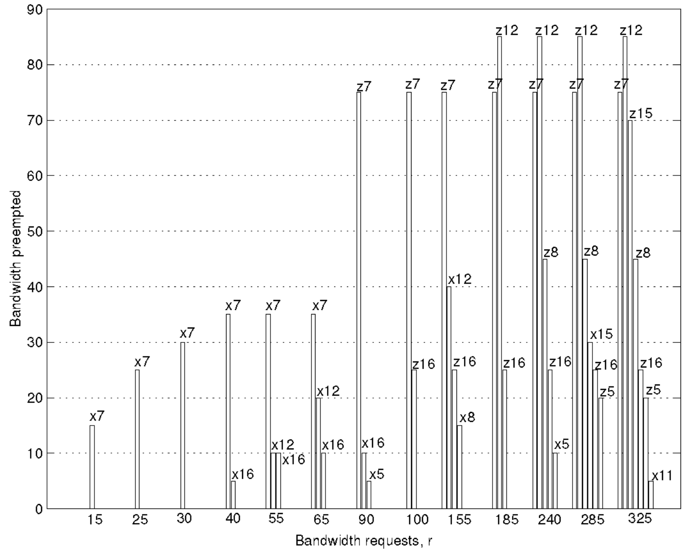

Fig. 11. Rate reduction and preemption for several values of r.

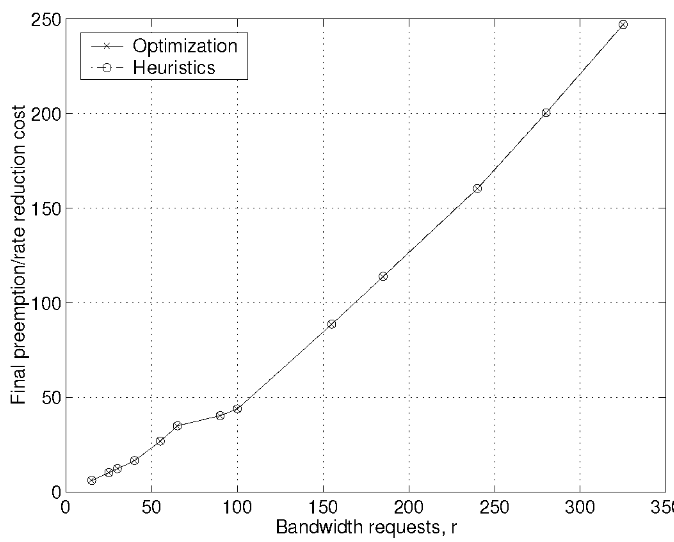

Fig. 12. Optimization and heuristic for adaptive preemption policy.

To illustrate the case in which r is larger than total bandwidth reserved by LSPs in $\mathcal{L}'$, suppose a request for $r = 600$ Mbps arrives. We observe that $\sum_{i=1}^{11} b(i) = 565$, which is less than r. In this case, following the flowchart in Fig. 9 we run the algorithm **Preemption** and keep selecting LSPs to be preempted until $r-$*preempt* is less than the bandwidth for the remaining LSPs in $\mathcal{L}'$. This means that traffic trunks l_3, l_7, l_9, l_{12}, l_{13}, and l_{15} are preempted, resulting in *preempt* = 440 Mbps, using the heuristic in Fig. 3. The remaining bandwidth, $r-$*preempt* = 160, is now suitable for our algorithm **Prep_Reduce** in Fig. 10. Running this heuristic, we realize that LSPs l_4, l_5, l_8, l_{11}, and l_{16} are preempted completely, making *preempt* = 595, and LSP l_{14} reduces its rate by 5 Mbps, which results in a total of exactly 600 Mbps available bandwidth for the new LSP setup request.

V. Performance Evaluation

In order to highlight the benefits of a preemption enabled scenario, we grouped the priority levels into three categories: low ($p = 5, 6, 7$); medium ($p = 2, 3, 4$); and high ($p = 0, 1$). We perform a simulation in which LSP setup requests were generated randomly in time (average 1.5 per second), with random bandwidth request b (varying from 1 to 10 Mbps), random priority level (0-7), and exponentially distributed holding time with average 500s. The total link capacity was 155 Mbps. We observed the probability of a successful LSP setup for low, medium, and high level priorities for preemption-enabled and non-preemptive approaches as well as the probability of rerouting. **Prept_Priority** is a commercial router preemption approach, that only considers the priority of the LSPs.

Fig. 13 illustrates the results obtained for each priority category. For the non-preemptive approach, the probability of LSP setup does not change with priority level. However, in a preemptive approach, low priority LSPs will be rerouted, while high priority LSPs will be more stable, always using the best path and only being rerouted in case of link failure. LSP setup probability is the same for **Preemption** and **Prep_Reduce** heuristic due to the fact that in the worst case **Prep_Reduce** will also preempt the whole LSP.

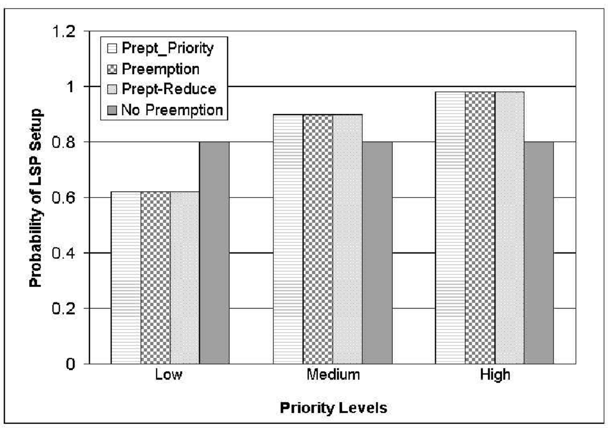

Fig. 13. LSP setup probability for preemptive and non-preemptive scenarios.

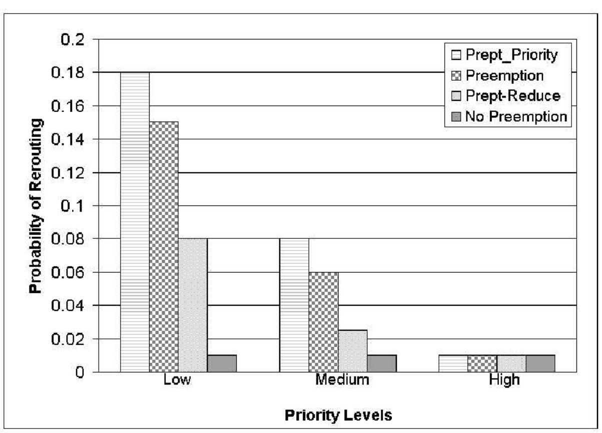

Fig. 14. LSP rerouting probability for preemptive and non-preemptive scenarios.

For a non-preemptive approach rerouting will only happen when link failure occurs, and we denoted a probability of 0.01 for that event. Fig. 14 shows the rerouting probability. For lower priority and medium priority LSPs, the rerouting probability is higher. For high priority traffic, the probability is almost the same as for non-preemptive approach, since this kind of traffic will not be preempted by other traffics. The rerouting probability for our **Prep_Reduce** heuristic is smaller for medium and low priority traffic, and would depend on the request r, since the LSPs would have their rate reduced to fit the new request, which implies less preemption and consequent rerouting.

In Fig. 15, both algorithms are compared by calculating a cost for each solution for each request r, using the same cost definition: Policy_cost = $\sum \mathbf{b} \cdot \mathbf{y} + \sum \mathbf{z} + b$, where b is the bandwidth requested by the new LSP, and $\mathbf{b}$, $\mathbf{y}$, and $\mathbf{z}$ are the same vectors defined earlier in this paper. The final cost achieved by the preemption policy complemented by the adaptive rate scheme is significantly smaller than the one obtained by the preemption policy by itself. Moreover, signaling costs are reduced due to the fact that rerouting is performed less frequently.

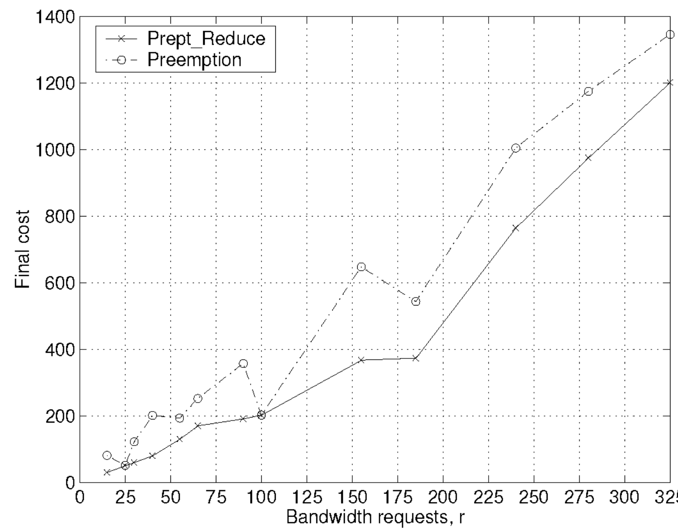

Fig. 15. Cost comparison between preemption policy and adaptive rate preemption policy.

VI. Conclusions

In this paper, a new preemption policy was proposed. The policy is complemented by an adaptive scheme that selects LSPs with lower priority and reduces their rate, up to a maximum percentage, in order to accommodate a new high-priority LSP setup request. The preemption policy combines the three main optimization criteria: number of LSPs to be preempted, priority of LSPs to be preempted, and amount of bandwidth to be preempted. Heuristics for both preemption and adaptive preemption policies are derived and their accuracy is demonstrated by simulation results. Performance comparisons of a non-preemptive approach, our preemption policy, the adaptive rate policy, and the policy in use by commercial routers show the advantages of using our preemption policy in a differentiated services environment. The proposed adaptive rate policy performs much better than the stand-alone preemption policy.

VII. Acknowledgements

The authors would like to thank the reviewer's valuable comments and suggestions. This work was supported in part by NASA Goddard and CAPES (The Brazilian Ministry of Education Agency).

References

[1] D. O. Awduche et al., *Requirements for Traffic Engineering over MPLS*, IETF, RFC 2702, September 1999.

[2] E. Rosen et al., *Multiprotocol Label Switching Architecture*, IETF, RFC 3031, January 2001.

[3] D. O. Awduche et al., *RSVP-TE: Extensions to RSVP for LSP Tunnels*, IETF, Internet Draft, August 2000.

[4] P. Vaananen and R. Ravikanth, *Framework for Traffic Management in MPLS Networks*, IETF Internet Draft, March 1998.

[5] A. Bobbio, A. Puliafito, and M. Tekel, "A Modeling Framework to Implement Preemption Policies in Non-Markovian SPNs," *IEEE Transactions on Software Engineering*, vol. 26, no. 1, pp. 36–54, January 2000.

[6] C.-G. Lee et al., "Bounding Cache-Related Preemption Delay for Real-Time Systems," *IEEE Transactions on Software Engineering*, vol. 27, no. 9, pp. 805–826, September 2001.

[7] J. Wang, Q.-A. Zeng, and D. P. Agrawal, "Performance Analysis of Preemptive Handoff Scheme for Integrated Wireless Mobile Networks," in *Proceedings of IEEE GLOBECOM'01*, San Antonio, TX, 2001, pp. 3277–3281.

[8] S. Poretsky, "Connection Precedence and Preemption in Military Asynchronous Transfer Mode (ATM) Networks," in *Proceedings of MILCOM'98*, 1998, pp. 86–90.

[9] S. Poretsky and T. Gannon, "An Algorithm for Connection Precedence and Preemption in Asynchronous Transfer Mode (ATM) Networks," in *Proceedings of IEEE ICC'98*, 1998, pp. 299–303.

[10] F. Le Faucheur et al., *Requirements for Support of Diff-Serv-Aware MPLS Traffic Engineering*, IETF Internet Draft, June 2001.

[11] J. Boyle, *Accomplishing DiffServ TE Needs with Current Specifications*, IETF Internet Draft, July 2001.

[12] M. Peyravian and A. D. Kshemkalyani, "Decentralized Network Connection Preemption Algorithms," *Computer Networks and ISDN Systems*, vol. 30, no. 11, pp. 1029–1043, June 1998.

[13] J. A. Garay and I. S. Gopal, "Call Preemption in Communication Networks," in *Proceedings of IEEE INFOCOM'92*, 1992, pp. 1043–1050.

[14] G. Armitage, *Quality of Service in IP Networks*, Macmillan Technical Publishing, Technology Series, 2000.