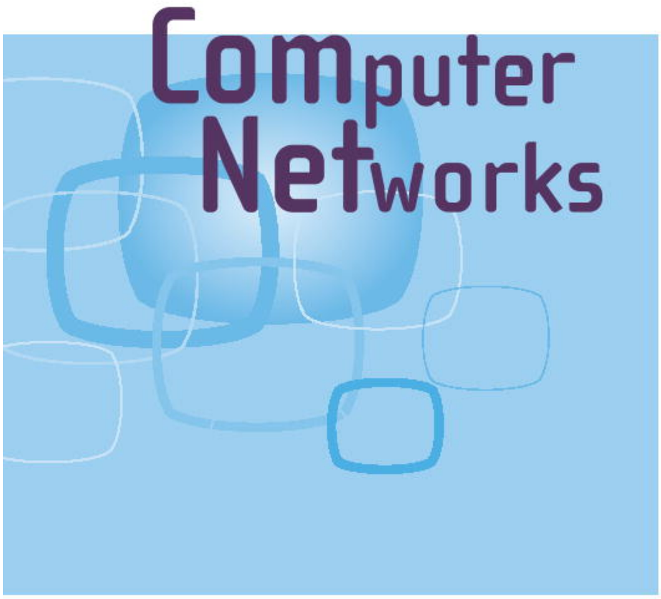
Volume 51 Issue 10 11 July 2007 ISSN 1389-1286



Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 2738–2752
Computer Networks
www.elsevier.com/locate/comnet

A framework for admission control and path allocation in DiffServ networks ☆

M. Ajmone Marsan, C. Casetti, G. Mardente, M. Mellia *

Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi, 24, I-10129 Torino, Italy

Received 23 March 2006; received in revised form 16 October 2006; accepted 16 November 2006
Available online 10 January 2007

Responsible Editor: Jaudelice Cavalcante de Oliveira

Abstract

We consider a Differentiated Service Domain, in which the domain administrator has to decide if to accept or to reject Bandwidth Reservation Requests (BRRs) requested by users. We first define an analytical approach and a methodology to determine the set of SLAs that can be effectively supported by a DiffServ IP network. We consider the Assured Forwarding Per Hop Behavior, and, based on the BRR probabilistic description, we derive a worst–case mathematical formulation for the overbooking probability, i.e., the probability that the traffic crossing any link of a source-destination path exceeds the link capacity. Next, we focus our attention to the problem of routing traffic arising from BRRs, i.e., the selection of paths along which traffic may flow. In particular, we show that the construction of an optimal set of paths is equivalent to the construction of a multicast tree, or a *Steiner Tree*, which is know to be an NP-hard problem. We therefore propose a class of simple heuristics, whose performance are assessed by simulations. Results show the effectiveness of the admission control criterium proposed, and that it is possible to increase up to 40% the amount of capacity a network provider can reserve to BRRs without violating the QoS constraints or to reduce the BRR blocking probability by a order of magnitude by using the proposed optimization algorithm.

Keywords: DiffServ; Admission control; Routing; QoS

1. Introduction

Traffic patterns on today's Internet have become more and more unpredictable, shifting from the ubiquitous client-server paradigm of the early days of the World Wide Web, to the peer-to-peer frenzy of the past few years. As if predicting user traffic were not sufficiently demanding, the introduction of a wide range of mobile services over the Internet is bound to give service providers quite a few headaches too. For these reasons, adequate tools to

☆ This work was supported by the European Community IST Network of Excellence EuroNGI. A preliminary version of this paper appeared in IEEE Globecom 2002 [4].
* Corresponding author. Tel.: +39 011 5644173; fax: +39 011 5644099.
E-mail addresses: ajmone@mail.tlc.polito.it (M.A. Marsan), casetti@mail.tlc.polito.it (C. Casetti), mardente@mail.tlc.polito.it (G. Mardente), mellia@mail.tlc.polito.it (M. Mellia).

1389-1286/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2006.11.022

support the provision of Quality of Service (QoS) guarantees to end users are sorely needed. Among them, the DiffServ [1] architecture is frequently touted as the cure-for-all solution.

The DiffServ architecture works by providing packet classification at the network ingress, and different treatment within the network according to a (small) set of classes, named Per-Hop Behaviors (PHBs). The various PHBs define a rich toolbox for differential packet handling by individual IP routers. In the DiffServ framework, a service contract, or Service Level Agreement (SLA), is established between a customer and a service provider, to specify the forwarding service that a customer should receive. An SLA encompasses several aspects of network communication, namely peak/average bandwidth guarantees, service outage duration and frequency. In this work however, we focus on the problem of determining whether enough resources are present in the network so that the bandwidth requirements of the novel and the already established SLAs are fulfilled. We therefore define a framework for Bandwidth Reservation Requests (BRRs) and BRR admission control procedures and algorithms. Once successful established, the service contract, though, does not discriminate among packet destinations, (or sources if they are being received by the customer). For this reason, it is of paramount importance that a service provider be capable of predicting, as it were, which parts of its core network are likely to become overloaded as a new BRR is accepted, and take appropriate actions. A preemptive solution is to route traffic from a new BRR over a set of paths that, at the same time, satisfy the user's bandwidth requirements while making sure that the bandwidth is evenly utilized across the whole domain. This paper addresses the issue of where to route traffic from new BRRs, once they are admitted into a provider's network. We stress that the solutions outlined in this paper are amenable to on-line and off-line implementation by a service provider.

Before tackling the problem of BBR admission and routing, it seems useful to recall the cornerstones of the DiffServ architecture. Traffic offered by the user is metered at its ingress node according to the user's traffic profile, and packets are marked accordingly so that they experience different PHBs. Common DiffServ PHBs are Expedited Forwarding (EF) [2], Assured Forwarding (AF) [3], and classic Best Effort (BE). The purpose of the EF PHB is to carry traffic from endpoints as if it traveled over a "virtual leased line"; on it, deterministic guarantees are offered. Provisioning the network for EF traffic is often accomplished by giving it strict priority over traffic marked by other code points. The AF PHB instead offers a soft guarantee: within each AF class, IP packets are marked with one of (two or three) possible drop precedence values. In case of congestion within a node, it tries to protect packets with better service profile by preferably discarding those with a higher drop precedence value. The AF PHB can be used in a point-to-point setting, as well as in point-to-multipoint configurations, where traffic can flow to different destinations (or come from different sources) at the same time.

The DiffServ guarantee paradigm is now supported in a number of IP routers of different make; still, not many network operators are yet exploiting DiffServ to offer guarantees to their customers, in spite of the potential increases in revenues. This fact is due in part to the difficulty in setting the parameters of the algorithms used for the differentiation of the treatment offered to the different traffic classes, but even more to the difficulty in determining how to implement a BRR Admission Control (BRR-AC) algorithm so as to protect the network from overloads, thus allowing the network to meet the bandwidth guarantees specified in the SLA contract.

In this paper we first tackle the BRR-AC problem by developing an analytical model to characterize the admissibility of a set of BRRs; this analytical model can also be used at run time to decide about the admission of a new BRR in a given network scenario. The implementation of the admissibility decision requires the global knowledge about the bandwidth allocated to currently active SLAs and the routes used by the corresponding traffic; thus, it requires the existence of a (centralized or distributed) server in the network, where BRR-AC decisions are taken, coherently with the Bandwidth Broker concept.

The model used to decide about BRR admissibility is based on a statistical approach, and can only approximately predict whether the acceptance of a set of BRRs allows the fulfillment of their requirements. However, it must be noted that, to the best of our knowledge, no previous approach exists in the literature to estimate the admissibility of a set of BRRs in a DiffServ scenario.

Then we propose a heuristic algorithm that tries to increase the effectiveness of the admission algorithm through a careful selection of routes from sets of sources to sets of destinations. Those BRRs that were accepted, and for which a route was selected, are "pinned" to the links of the route. Commonly, this is achieved using MPLS [5,6] inside the core network.

2. Related works

The main conceptual difference between classic admission control [7] problems and the AF-BRR admission control problem in a DiffServ network consists in the BRR definition itself. Indeed, in a AF-BRR, the SLA contracted between the network and the user involves in general more than one source/destination pairs. In particular, two possible types of BRR are possible: the first refers to traffic generated by a single user and directed to (possibly) multiple destinations, while the second considers traffic going from (possibly) multiple sources to a single user. Each BRR is described in terms of parameters, such as assured bandwidth going to (coming from) a set of possible destinations (sources) within the same DiffServ Domain, or, possibly, egress (ingress) nodes connected to other domains. The choice of the links which will then be used to carry the traffic belonging to a given AF-BRR is therefore different from the classic routing problem in a point-to-point requests. It indeed can be formalized as finding the set of paths which will be used to route a new BRR, so that the amount of bandwidth available to all AF-BRRs over such set of paths is guaranteed. This problem is equivalent to the well-known Steiner Tree problem [8].

The problem of finding the best Steiner tree has also been faced in the optimization of Virtual Private Networks, and it has been traditionally named in the literature as the "hose model" [12]. The hose model for bandwidth reservations has been proposed in the context of VPN (virtual private network) provisioning. A VPN is an emulation of services provided by Private Networks (PN) – leased lines connecting a set of sites. With an increasing number of VPN endpoints, an alternative to the standard point-to-point (or customer-pipe) approach has been proposed in the hose model: given a network with link capacities and a set of VPN sources/destinations, one needs to know an upper limit of the amount of traffic entering and leaving each node from and to the other nodes. Then, a VPN hose is created connecting all nodes, with preference for sharing as many links as possible. The structure of the hose can be a tree, or any other connected graph. Once the set of routes (or the Steiner Tree, or the hose) has been defined, classic approaches to test if the network has enough resource to allocate to the novel BRR can be applied. If not enough resources are available, then the BRR will be rejected.

In the literature several Measurement Based Admission Controls have been proposed, but also parameter-based approach are pursued. End-point admission control through probing [13] is the most interesting approach. Probes are sent out in the requested traffic class and the results gathered by the egress router in terms of packet loss, jitter, transmission time etc. The egress router itself decides whether the flow can be accepted. This has the problem of bandwidth stealing, since the probes do not show the effect of accepting the new traffic on already accepted BRRs. Also there is an overhead in the form of large setup time.

More traditional approaches comprise: pricing mechanisms, which are proposed as a viable means to the BRR admission control in [14] for example, backward learning algorithms [15], and measured based mechanisms coupled with service curve approaches [16].

3. Admission control for BRRs in DiffServ networks

The BRR admission problem requires the following input information:

- the (logical) *topology* of a single DiffServ Domain, comprising M nodes and L links; each link $l, 1 \leqslant l \leqslant L$ is supposed to reserve capacity C_l to the AF class of service. Each node m can be classified as either ingress/egress node if users (i.e., non-specific traffic generators) are connected to that particular node, or *core* node, if it is a pure transit node, i.e., no traffic is either generated or directed to that node;
- the *routing algorithm* used on the logical topology, which is assumed to be known (for example, Shortest Path); if routing identifies multiple paths, load balancing among different paths is allowed;
- the set of *users* that request service from the network; each user is attached to an ingress/egress node, and can request more than one service from the network, i.e., multiple BRRs can refer to the same user;
- a *definition of the BRR*s of interest; in particular, we consider two possible types of BRR: the first refers to traffic generated by a single user and directed to (possibly) multiple destinations, while the second considers traffic going from (possibly)

multiple sources to a single user. Each BRR is described in terms of assured bandwidth going to (coming from) a set of possible destinations (sources) within the same DiffServ Domain, or, possibly, egress (ingress) nodes connected to other domains. The BRRs demand statistical bandwidth guarantees on traffic, i.e., the traffic transmitted (received) by the source (destination) subscribing to the SLA; therefore, each BRR is characterized a statistical bandwidth guarantees;

- a probabilistic *description of the traffic* associated with every BRR, in terms of fraction of overall traffic going from a user to the set of its destinations, or coming from the set of possible sources to the user, i.e., an average traffic matrix. Such probabilstic description is compatible with the provision of soft guarantees.

The BRR admission problem output is the acceptance region for BRRs, defined as a set of inequalities that must be verified in order to guarantee that the network fulfills BRR requests.

4. Notation

The following notation is used throughout this paper:

- C_l is the capacity of link l that is reserved to the AF class of service;
- BRR_i^k is the k-th BRR subscribed to by user i;
- B_i^k is the bandwidth requested from by i as specified in BRR_i^k; B_i^k can refer to either the outgoing assured bandwidth allocated to user i to transmit to a set of possible egress nodes, or the incoming assured bandwidth that user i is allocated to receive from a set of possible ingress nodes;
- Π_i^k is the probability with which BRR_i^k must be guaranteed by the network, i.e., the probability that user i is allowed to send (receive) traffic using a bandwidth equal to or larger than B_i^k;
- $r_{i,j}^k$ is the probability with which the traffic belonging to BRR_i^k is directed toward (coming from) the egress (ingress) node j; since the destination (source) of traffic belonging to the same BRR can be possibly more than one egress (ingress) node, the notion of traffic matrix associated with each BRR is required; obviously $\sum_j r_{i,j}^k = 1$; we assume that a route toward (coming from) an egress (ingress) node is always available, regardless of whether there actually is traffic flowing to (from) that node;
- $pdf(\Phi_{i,j}^k, \omega)$ is the pdf of the random variable $\Phi_{i,j}^k$, which characterizes the AF traffic referring to BRR_i^k and egress router j. Note that $\Phi_{i,j}^k$ is not related to the current traffic user i is sending into the network, rather to the possible traffic the user might send within its BRR definitions.

5. BRR admission control

The methodology we propose has two goals: the first is to obtain a probabilistic description of the AF link load through the knowledge of the probabilistic description of traffic patterns and the paths each flow follows in the network. The second goal is to derive a statistical description of the overbooking probability for every BRR, in order to define the maximum bandwidth that can be guaranteed to each BRR. This identifies an *acceptance region* including the sets of BRRs that can be guaranteed, given their contracted overbooking probability. Alternatively, this can be used as an *admissibility control criterion* for new BRR requests. The admission criterion is defined for each ingress or egress node of the DiffServ Domain, given the BRRs already accepted.

In order to evaluate the Assured Forwarding traffic the network must support, we need to know $pdf(\Phi_{i,j}^k, \omega)$ for all i,j,k. These quantities are difficult to obtain from users, because in general they are not related to the real traffic user i is injecting into or receiving from the network. Indeed, a user can request a BRR characterized by a value B_i^k that in general can be higher than the peak bandwidth the user expects to need (in which case the user is *overprovisioning* its BRR, so that all of its traffic is marked with the contracted drop precedence value), or even smaller than the user's average bandwidth usage (in which case the user is *underprovisioning* its BRR, and a portion of user traffic is marked with a higher drop precedence value, and receives a lower service level from the network). Thus, it is unrealistic to suppose $pdf(\Phi_{i,j}^k, \omega)$ to be known. This makes $pdf(\Phi_{i,j}^k, \omega)$ intrinsically hard to quantify, and impossible to derive from measures.

We can however conjecture what possible forms $pdf(\Phi_{i,j}^k, \omega)$ can take. $\Phi_{i,j}^k$ are limited-support random variables, as their values are upper-bounded by B_i^k. Thus

$$\omega \in \{0, B_i^k\} = I. \tag{1}$$

Moreover, for a given destination (source) node j, we have that

$$E[\Phi_{i,j}^k] = r_{i,j}^k B_i^k, \tag{2}$$

since the traffic pattern of each BRR is known. Finally, being $pdf(\Phi_{i,j}^{k},\omega)$ a density probability function, we have

$$\int_{I} pdf(\Phi_{i,j}^{k},\omega)\mathrm{d}\omega = 1. \tag{3}$$

However, there are infinite possible $pdf(\Phi_{i,j}^{k},\omega)$ that satisfy (1)–(3). Among those, we chose $pdf(\Phi_{i,j}^{k},\omega)$ as an on–off distribution, whose analytical expression can be written as

$$pdf(\Phi_{i,j}^{k},\omega) = (1-r_{i,j}^{k})\cdot\delta(\omega) + r_{i,j}^{k}\cdot\delta(\omega - B_{i,j}^{k}), \tag{4}$$

because this is the *pdf* that maximizes the variance, and thus the uncertainty associated with the AF traffic being carried by the network. Indeed, it is intuitive to see that the worst-case scenario is given by users who send (receive) an amount of traffic equal to their contracted bandwidth to (from) a single destination (source) at a time. This allows us to derive a worst-case scenario, hence a conservative methodology.

Given a link over which BRRs are setting up reservations, consider the overall traffic that is expected to flow on such link if all BRRs were accepted. Let $\alpha_{i,j}^{k,l}$ be a binary variable, i.e., either equal to 0 or to 1, that takes into account the routing of BRR_i^k; in particular,

$$\alpha_{i,j}^{k,l} = \begin{cases} 1 & \text{if } \mathrm{BRR}_i^k \quad \text{traffic to(from) } j \text{ crosses link } l \\ 0 & \text{otherwise.} \end{cases}$$

If link l belongs to more than one path that carries traffic belonging to the same BRR_i^k, then the percentage of AF traffic flowing on l is described by a random variable $\Psi_i^{k,l}$, whose average is given by

$$E[\Psi_i^{k,l}] = \mu_i^{k,l} = B_i^k \sum_j \alpha_{i,j}^{k,l} r_{i,j}^k \tag{5}$$

and then it is possible to derive an on–off $pdf(\Psi_i^{k,l},\omega)$ following the reasoning outlined by (1)–(3), and assuming an on–off distribution. The standard deviation associated with the total AF traffic belonging to BRR_i^k to (or from) egress j, and flowing on link l is then

$$\sigma_i^{k,l} = \sqrt{E[(\Psi_i^{k,l})^2] - E[\Psi_i^{k,l}]^2} = B_i^k\sqrt{(\mu_i^{k,l} - (\mu_i^{k,l})^2)}. \tag{6}$$

Given the statistical description of the AF traffic offered by BRRs, it is possible to derive a statistical description of the requested AF traffic flowing on links through the evaluation of the random variable ϕ^l, which is given by the convolution of all PDFs that describe the traffic from BRRs crossing link l.

In order to do that, we assume statistical independence among the $\Psi_i^{k,l}$. While this assumption holds if we are considering two different BRRs, it does not hold for traffic belonging to the same BRR, and flowing on different paths. Indeed, considering a link l, $\Psi_i^{k,l}$ can either be equal to 0 or to B_i^k, since only one destination j at a time can have $\Phi_{i,j}^k = B_j^k$; therefore, for a given source i and BRR k, there is statistical dependence among $\Psi_i^{k,l}$ on links belonging to disjoint paths leading to different destinations j. However, the independence assumption is a worst-case scenario, because it could lead to the case where the BRR traffic is flowing on multiple paths at the same time, thus causing the total amount of traffic outgoing (incoming) from (to) the same source (destination) node to exceed the contracted bandwidth. Under this assumption, we can write that

$$pdf(\phi^l,\omega) = \otimes_{k,i} pdf(\Psi_i^{k,l},\omega), \tag{7}$$

where $\otimes$ denotes the convolution operator.

Since the exact evaluation of the distribution of the reserved traffic on all links can be too expensive, we propose to approximate it by applying the Central Limit Theorem. The random variable describing the traffic reserved on a given link is the sum of several independent traffic variables, and thus tends to be normally distributed. Using the central limit theorem, we can therefore state that the average and the standard deviation of traffic reserved on link l are

$$\mu^l = E[\phi^l] = \sum_i \sum_{k\in\mathrm{BRR}} \mu_i^{k,l}, \tag{8}$$

$$\sigma^l = \sqrt{E[(\phi^l)^2] - \mu^{l2}} = \sqrt{\sum_i \sum_{k\in\mathrm{BRR}} (\sigma_i^{k,l})^2}, \tag{9}$$

where μ^l and σ^l are, respectively, the average and standard deviation of the amount of traffic flowing on link l.

The resulting distribution of traffic crossing link l is approximated by a Gaussian distribution and the probability that the traffic crossing link l is greater than C_l, i.e., the *link overbooking probability* P^l, is given by

$$P^l = \mathrm{Prob}\{\omega > C_l\} = \frac{1}{2}\cdot\mathrm{erfc}\left(\frac{C_l - \mu^l}{\sqrt{2}\cdot\sigma^l}\right). \tag{10}$$

For each BRR, it is possible to relate the probability P_i^k with which BRR_i^k cannot be guaranteed by the network to the link overbooking probability P^l

$$P_i^k = 1 - \sum_j r_{i,j}^k \prod_l \alpha_{i,j}^{k,l}(1 - P^l), \tag{11}$$

because there must be no overbooking on every link through which the traffic from source i, related to BRR_i^k, flows.

Thus, from (10) and (11) it is possible to derive a system of inequalities that, for each BRR, must hold to guarantee that no overbooking is present, and all the committed BRRs are satisfied

$$(1 - P_i^k) \geqslant \Pi_i^k, \quad \forall i, k. \tag{12}$$

The resulting system contains $|\{\mathrm{BRR}_i^k\}|$ inequalities, that completely specify the admissibility region of the BRRs. The system is non-linear, and must be solved using numerical techniques. The same system of inequalities can also be used as admissibility control criterion for new BRR requests coming to the network, in which case all inequalities must be verified to accept the new request, given a previous existing set of already admitted BRRs.

6. Path selection for BRRs

Usually linked to the admission control problem is the selection of paths that allow newly admitted BRRs to be routed without interfering with path allocations of existing BRRs (i.e., without causing the violation of QoS guarantees already provided to existing BRRs). The problem of finding the set of paths that will be used to transport traffic from a newly admitted BRR is formalized using a graph theory approach. The topology of the DiffServ domain is modeled by a Directed Graph $\mathscr{D} = \{\mathscr{V}, \mathscr{A}\}$ in which $\mathscr{V}$ is the set of vertexes ($|\mathscr{V}| = M$) and $\mathscr{A}$ is the set of directed arcs ($|\mathscr{A}| = 2L$). A vertex represents a node in the topology, while two directed arcs $\langle i, j\rangle, \langle j, i\rangle \in \mathscr{A}$ between nodes $i, j \in \mathscr{V}$ represent a link. Each arc is weighted by two costs, (μ_{ij}, σ_{ij}), which represent the *average* and *standard deviation* of the traffic which is flowing on arc $\langle i,j\rangle$.

The set of nodes $\mathscr{D}_i^k = \{s \in V | r_{ij}^k \neq 0, j \neq i\} \subset V$ includes all destination (source) nodes of BRR_i^k. The cardinality of $\mathscr{D}_i^k$ will be indicated by D_i^k. Then, given a set of already accepted and routed BRRs, it is possible to derive for each arc the cost (μ^l, σ^l).

The routing optimization problem can then be formalized as finding the set of arcs T_i^k which will be used to route a new BRR request, so that the maximum average overbooking probability experienced by all $\{\mathrm{BRR}_i^k\}$ over such set of arcs, indicated as $P_i^k(T_i^k)$, is minimized, i.e., $\min(\max_{i,k} P_i^k(T_i^k))$. If $P_i^k(T_i^k) \leqslant (1 - \Pi_i^k)$, then the new BRR will be accepted, otherwise it will be blocked.

This problem is equivalent to the well-known *Steiner Tree problem* [8], which can be summarized as the problem of finding the minimum cost tree T_i^k that connects a source node i to a subset of vertexes in digraph $\mathscr{D}$. As cost function, the overbooking probability is considered, which unfortunately is a non-linear function of (μ_l, σ_l), transforming the formulation in a non-linear problem. Moreover, also in its original formulation, the Steiner Tree problem is known to be NP-complete, and therefore can only be solved using heuristics.

6.1. Proposed heuristic

The limited amount of time that can be devoted to solve the problem, i.e., to reply to a user's BRR, suggests that the Steiner Tree problem is best solved using heuristics with limited complexity. We therefore propose a simple heuristic, whose complexity is very limited and depends on two tunable parameters. In particular, the construction of the Steiner tree is obtained as union of pre-computed paths according to any given metric, each one connecting the source node s to a particular destination d. An iterative algorithm is then used to compute several trees, and select among them the one which minimizes the maximum overbooking probability $P_i^k(T_i^k), \quad \forall i, k$.

Let $\{P_{s\to d}^i, i = 1, 2, \ldots, K\}$ be an generic ordered set of K paths pre-computed between s and d using a metric common to all of them. In the following, the metric of choice is the number of hops, so that path number 1 is the shortest path.[1] Let $T_i^k(n)$ be the solution to the Steiner tree problem obtained at iteration n. For each destination d, a single path, identified as opt, is selected, and $T_i^k(n)$ is then obtained as the union of all the arcs of $P_{s\to d}^{\mathrm{opt}}$. Therefore different solutions are obtained by selecting different sets of paths.

The algorithm we propose tries to efficiently explore the state space of a possible set of paths, whose number grows as a combinatorial function

[1] In case less than K paths exist, only the available paths will be considered.

of K. Instead of considering all possible combinations of paths, at each iteration $K \cdot D_i^k$ different solutions are tested, each one obtained by simply changing one path at a time, i.e., for a given destination d, test all trees obtained by considering all K paths $P^i_{s \to d}$. This defines $K \cdot D_i^k$ "neighbors", i.e., solutions that differ from the previous one by a single path. At the end of the iteration, the best neighbor is selected. A maximum number of iterations Z is defined to limit the complexity of the algorithm. This algorithm fall in the class of the "Steepest Descend" class of metaheuristics according to the operative research naming.

Fig. 1 shows two possible trees (arcs included in the tree are thicker), obtained considering the source node s, and two destination nodes d_1, d_2. Two different paths to d_1 are considered, $P^i_{s \to d_1} = \langle s, 2\rangle, \langle 2, d_1\rangle$, and $P^j_{s \to d_1} = \langle s, 3\rangle, \langle 3, d_2\rangle, \langle d_2, d_1\rangle$, yielding two different trees.

Fig. 2 reports a formal description of the algorithm. Lines 1–7 build the initial solution as the union of all the first-selected paths for all destinations. Lines 10–24 then iterate Z times the construction of possible better solutions, by building for each destination (line 12) all possible trees considering all K paths from i to d (lines 14–22). opt[d] is used to store the best path found so far toward destination d.

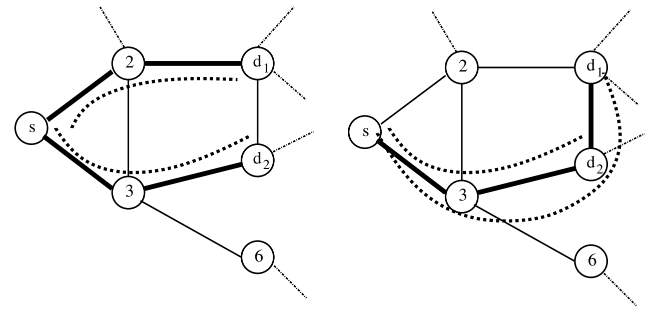


Fig. 1. Two possible trees from source node s to destinations nodes d_1 and d_2 obtained by considering $P^i_{s \to d_1} = \langle s, 2\rangle, \langle 2, d_1\rangle$, and $P^i_{s \to d_1} = \langle s, 3\rangle, \langle 3, d_2\rangle, \langle d_2, d_1\rangle$.

Considering as basic operation the evaluation of the minimum–maximum overbooking probability for all BRR, the algorithm complexity is $O(ZMK)$, as at most Z iterations are possible, each of which requires to consider for all destinations $D_i = O(M)$ at most K paths. The computation of K paths is done offline and therefore its complexity does not affect the time required to reply to a BRR. It can be obtained using variations of the Dijkstra's algorithm with computational complexity $O(KL \log M)$. In particular, in our implementation of the algo-

1. $T_s(0) = \emptyset,\ n = 0$
2. // Build the initial solution
3. forall $(d \in \mathcal{D}_s)$
4. {
5. $\quad T_s(0) = T_s(0) \bigcup \{< l, m > \in P^1_{s \to d}\}$
6. $\quad opt[d] = 1$
7. }
8. $T_s^* = T_s(0)$
9. // Iterate to build other solutions
10. for $(n = 1;\ n \le Z;\ n{+}{+})$
11. {
12. $\quad$ forall $(d \in \mathcal{D}_s)$
13. $\quad$ {
14. $\qquad$ for $(i = 1;\ i \le K;\ i{+}{+})$
15. $\qquad$ {
16. $\qquad\quad T_s(n) = \emptyset$
17. $\qquad\quad T_s(n) = T_s(n) \bigcup \{< l, m > \in P^i_{s \to d}\}$
18. $\qquad\quad$ forall $(j \in \mathcal{D}_s \setminus \{d\})$
19. $\qquad\qquad T_s(n) = T_s(n) \bigcup \{< l, m > \in P^{opt[j]}_{s \to j}\}$
20. $\qquad\quad$ if $(\max_s(p_s(T_s(n))) < \max_s(p_s(T_s^*)))$ then
21. $\qquad\qquad T_s^* = T_s(n); opt[d] = i$
22. $\qquad$ }
23. $\quad$ }
24. }
25. return T_s^*

Fig. 2. The heuristic algorithm.

rithm, the set of paths $\{P^i_{s\to d}\}$ is chosen as the set of K shortest paths from source node s to destination node d.

7. Performance results

We first present results to assess the performance of the admission control criteria. To optimize the routing of BRR, we set $K = 2$ and $Z = 10$. A discussion on the impact of those parameters will be presented in Section 7.5.

A software tool was developed to obtain experimental results for the proposed methodology. Given a description of the problem, including the BRR definitions in terms of $\{B^k_i, r^k_{i,j}, \Pi^k_i\}$, the tool can evaluate the system of inequalities defined in (12), and check whether the BRR set is admissible or the network is unable to fulfill its commitment to the contracted SLAs.

The tool also finds a set of BRR committed bandwidths $\{B^k_i\}$ for which all the inequalities in (12) hold, but no B^k_i can be increased without violating at least one inequality. We call this a *border* point, which indicates that the AF network capacity is completely exploited, and no already present BRR can ask for larger bandwidth without violating the guarantees of at least one BRR. Given that it is impossible to solve the system (12) in closed form, the tool iteratively seeks a border solution, trying to increase all B^k_i up to a value in which it is not possible to further increase any of them. Different "increase" algorithms can be applied, so that more than one border point can be found from a starting BRR set. In this paper, we set the initial $\{B^k_i\}$ to be 1/1000 of link capacity, then iteratively pick a BRR at random, and try to increase its committed bandwidth by a small fraction (around 1% of $\{B^k_i\}$). The algorithm stops when no increase can be applied for 100 consecutive iterations.

To test the performance of the proposed approach, we also developed an event-driven simulator based on the ANCLES tool [9]. This is a connection-oriented simulation tool, which can measure the performance of a max–min-fair network that carries elastic traffic connections. The simulator takes the network description as input, as well as the BRR set defined according to the pair $(B^k_i, r^k_{i,j})$. It then simulates the connection generation process at user level: during the simulation, each user requests that connections to a destination are set up, according to the traffic pattern defined by its BRR^k_i. Each connection is associated with a given (random) amount of data to be transferred, so that the average offered load is equal to B^k_i. The new connection is then routed over the path chosen by the heuristic toward the destination, and the instantaneous amount of bandwidth is evaluated according to a max–min-fair share algorithm, i.e., the bottleneck capacity is uniformly split among all connections flowing through the bottleneck channel at each time instant. Finally, the connection is terminated after all user data have been transferred.

During the simulation, the tool evaluates the probability that the guarantees to BRR^k_i are not met, i.e., it computes the *measured* overbooking probability $\widehat{P}^k_i$. This overbooking measure can then be compared to the one predicted by the proposed methodology.

As a first test, we consider a single DiffServ domain, comprising $M = 32$ nodes and $L = 138$ links, arranged in a randomly generated topology.[2] The GT-ITM [11] tool was used to generated the topology. Each link has the same capacity $C_l = 1$; such capacity is completely devoted to the AF service. Thirty-two users are present, one for each node, so that all nodes are ingress/egress nodes. Each user generates BRRs for three different destination sets, so that a total number of 96 BRRs are present. To simplify the scenario, we suppose that all BRRs are identical. In particular, we consider only BRRs where users are traffic sources, and traffic is routed uniformly to d destinations, i.e., for each BRR, d destination nodes are selected at random for which $r^k_{i,j} = 1/d$, while the remaining M–d-1 nodes do not receive connection requests from that particular BRR user.

7.1. Comparison with a persistent traffic model

In this test, BRR^k_i comprises N simultaneous connections, each requesting a bandwidth equal to $\frac{B^k_i}{N}$, so that the total traffic offered by BRR^k_i is equal to B^k_i. Each connection randomly selects one possible destination, uniformly chosen among the $d = 31$ egress points. Each connection must transfer an average amount of data corresponding to 100 KB, and as soon as a connection ends, another request immediately follows, so that the total offered traffic related to BRR^k_i is always constant and equal to B^k_i. If the bandwidth available to a

[2] Different random topologies were tested, without observing major differences from the one presented in this paper.

BRR is throttled because of congestion in the network, the duration of the connection grows longer. While this is not a realistic traffic model, it is very similar to the one adopted in the analytical methodology, where $pdf(\Phi^k_{i,j}, \omega)$ is modeled by an on–off distribution.

The overbooking probability is computed as the ratio between the starved traffic and the contracted traffic. By "starved traffic" we indicate traffic that obtained less bandwidth than requested.

In the scenario presented here, the BRR set represents a possible border point, determined such that $\Pi^k_i \geqslant 0.9$ for all BRRs. The resulting scenario was simulated in order to evaluate the accuracy of the prediction on the overbooking probability.

Fig. 3 shows simulation results and our analytical results.

For each BRR, the figure reports both the simulated and the computed overbooking probability P^k_i. The computed probability is plotted by the solid line, while the simulated probabilities are plotted for three separate cases, referring to different numbers of connections, N, by which each BRR is represented. For ease of visualization, the BRRs are sorted in decreasing order of estimated overbooking probability.

Considering the estimated overbooking probability, it must be noted that not all BRRs show $P^k_i = 0.1$, even if the considered scenario is a border point: only the leftmost BRRs show an estimated overbooking probability that is equal to 0.1, nonetheless it is not possible to increase other BRR committed bandwidth without at the same time violating at least one of these BRR constraints. This is not surprising, since we are considering a scenario

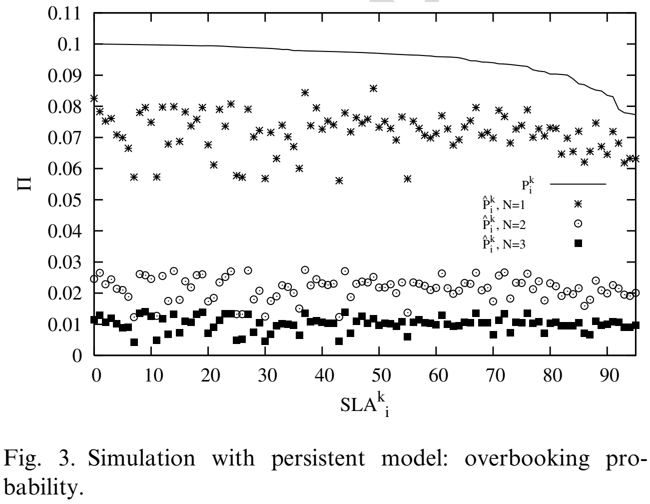


Fig. 3. Simulation with persistent model: overbooking probability.

where each BRR involves all possible destinations, and thus probably most of the links in the topology.

Considering the measured overbooking probability when $N = 1$, we can observe that $\widehat{P}^k_i < P^k_i$. This was expected, since the analytical model provides an upper bound to the actual overbooking probability. The fact that there is a difference between the simulation result and the analytical prediction is due to the assumption of statistical independence among the $\Psi^{k,l}_i$ (used in the model, but obviously violated in the simulations), which is equivalent to consider, during the computation of ϕ^l, that the traffic coming from any BRR may grow up to B^k_i for all destinations, without taking into account that the total bandwidth originating from one user cannot exceed B^k_i. If we consider the other cases, we instead observe the effect of the assumption that each $\Psi^{k,l}_i$ is modeled by an on–off process, where the bandwidth value within on periods is B^k_i, while in the simulations the bandwidth used by each BRR toward any destination can assume values $\{j(B^k_i/N)\}^N_{j=0}$.

7.2. Comparison with Poisson traffic model

We now consider a more realistic traffic model; the traffic offered by BRR^k_i is modeled by connection requests that arrive according to a Poisson process, in such a way that the average offered traffic is equal to B^k_i. Each BRR can have an average number of active connections equal to N. Being this a statistical model, there can be periods of time during which the network is overloaded, as well as periods of light utilization. In this scenario, the overbooking probability is defined as the ratio between the time during which the user was starved, and the simulation time (ignoring the initial transient phase).

We are interested in seeing if a border scenario which is admissible according to the introduced methodology is still admissible under the Poisson traffic assumption.

Fig. 4 reports both the estimated and the measured overbooking probabilities, where, except for the traffic model, the scenario is the same as considered in the previous case. We reported the cases $N = 1,2,3$, $N = 1$ being rather unrealistic. Our aim focus went on how the statistical multiplexing introduced by the Poisson model increases the gap between our worst-case estimate and the measured overbooking probabilities. Indeed, we see that $\widehat{P}^k_i < P^k_i$ holds for all BRRs and the gap becomes more and more noticeable as N increases. This is

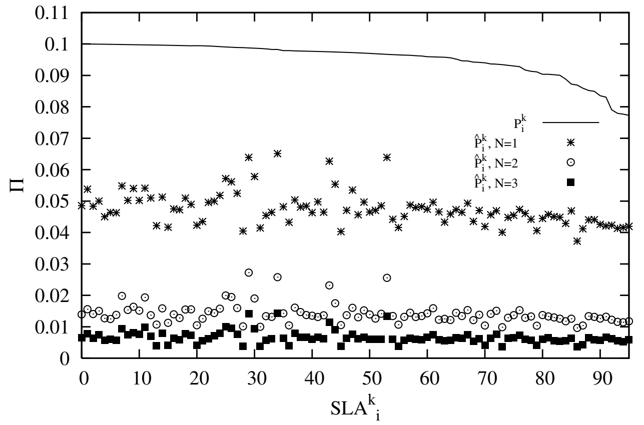


Fig. 4. Simulation with Poisson model: overbooking probability.

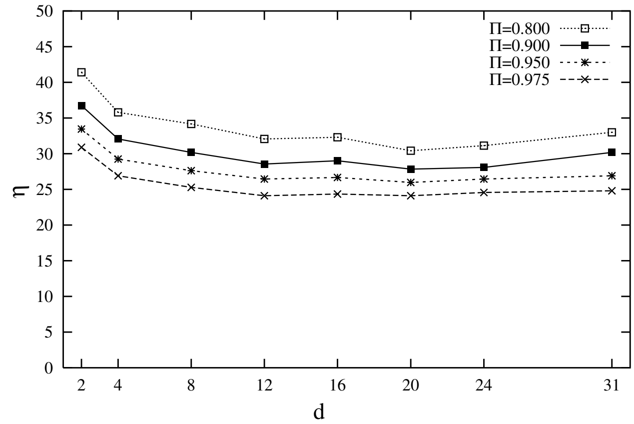


Fig. 5. Statistical gain versus *d*.

due to the large granularity introduced by the assumption that each connection generated by a BRR user request has bandwidth equal to B_i^k/N. Thus, every time instant when there are more than N connections from the same user, is a time when the current load is higher than the contracted load. This induces congestion on all the paths where connections are routed, resulting in an increased overbooking probability for all the BRRs sharing links on such paths. However, as N increases, the granularity decreases and the chance that the "excess" connections all contribute to violating the contracted load on the same link become slimmer.

7.3. Statistical gain

In this section we quantify the multiplexing gain allowed by the proposed framework. In particular, we quantify the portion of the network bandwidth that can be devoted to AF BRRs by a network operator (an ISP). Let η be the ratio (in %) between the committed BRR bandwidth and the weighted aggregate network bandwidth (sum of the bandwidths of all links over the average minimum distance among nodes). i.e.,

$$\eta = \frac{\sum_{i,k} B_i^k}{E[\Delta]\sum_l C_l} \qquad E[\Delta] = \frac{\sum_{i,j\neq i}\Delta(i,j)}{N(N-1)}, \tag{13}$$

where $\Delta(i,j)$ is the minimum-hop path length between node i and j. η represents the percentage of network resources that can be allocated to users given a specific number of selected destinations and the committed overbooking probability. We consider the same network topology as before, and study the impact of some of the parameters that are used to define the BRRs.

To better understand the relationship between the committed BRR bandwidth and the possible scenarios an ISP can face, Fig. 5 outlines an interesting behavior of η, and underscores its dependence on the number of possible destinations where the BRR traffic can be routed. We compare different border scenarios by varying the number of possible destinations d of BRRs, and the values of the overbooking probability Π. Each point is obtained by averaging 10 different border scenarios. If the number of destinations is small, the number of traffic relations per link is small; each has a high average data rate, which translates in a smaller σ^l, i.e., in a more deterministic traffic pattern. This allows a larger amount of traffic allocation into the network. η progressively decreases as the number of destinations increases, only to increase again for large numbers of destinations, when the effect of statistical multiplexing takes over. Indeed, when the number of destinations is large, the amount of traffic routed to each destination becomes smaller, so that the standard deviation $\sigma_i^{k,l}$ in (6) becomes small.

Fig. 6 reports η as a function of the committed overbooking probability for values $d = 8,16,31$, and compares the predicted values of η with those that could have been achieved if Mean Bandwidth Allocation (MBA) were used, and if peak allocation ($\Pi = 1$) were chosen. In particular, the MBA scenario considers $pdf(\Phi_{i,j}^k, \omega) = r_{i,j}^k \cdot \delta(\omega - B_{i,j}^k)$, which is equivalent to assuming the traffic offered by each BRR to be perfectly known and deterministic. This indeed represents the maximum amount of bandwidth that can be committed to AF traffic with $P_i^k \neq 1$, and if more traffic is accepted, the overbooking probability is always equal to 1.

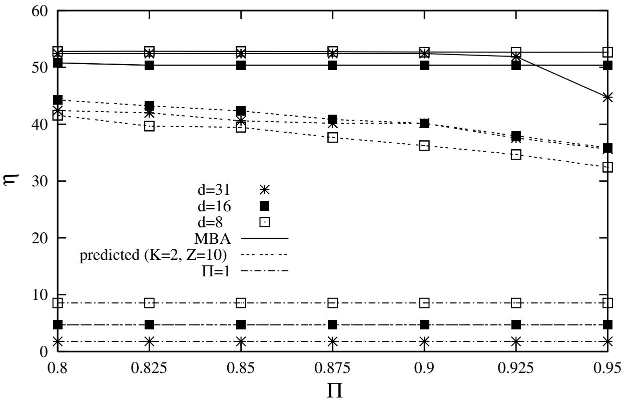


Fig. 6. Statistical gain versus Π.

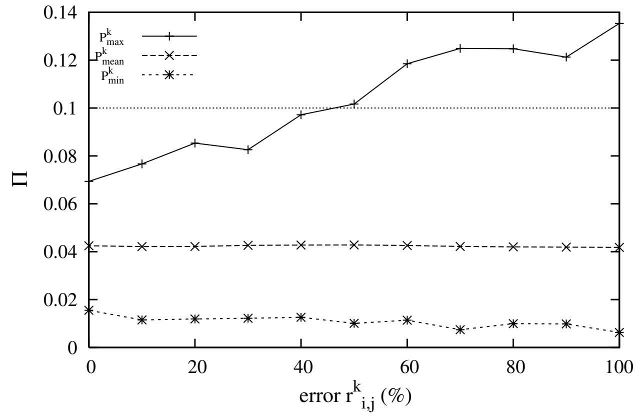


Fig. 7. Average, minimum and maximum overbooking probability experienced by BRRs for different percentage of uncertainty.

Fig. 6 shows that the proposed BRR-AC permits to commit to AF between 30% and 45% of the aggregate network bandwidth, depending on the overbooking probability contracted by the BRRs. This is about five times larger than peak allocation, and only about 10% points less than MBA allocation (which does not provide any bandwidth guarantee). This shows the effectiveness of the proposed methodology, that exploits network capacity while guaranteeing the overbooking probability. In addition, a slight decrease of the committed bandwidth is noticeable for large values of Π, as the stricter QoS required limits the multiplexing gain. Finally notice that $d = 8$ enforces a tighter limit on η, as the limited number of destinations imposes a more conservative bandwidth allocation; at the same time, the gain is reduced by the smaller amount of statistical multiplexing with respect to higher values of d.

7.4. Resilience to parameters uncertainty

We have shown so far that the BRR admission control criterion adopted correctly predicts the bandwidth availability perceived by users. It requires as input from the user a description of the total bandwidth requirement (B_i^k), and the traffic split among possible destinations (sources) ($r_{i,j}^k$). While the knowledge/prediction of B_i^k may be possible, the *perfect* knowledge of the traffic split may be very hard to obtain. Therefore we are interested in the impact of uncertainty in $r_{i,j}^k$ onto the admissibility criterion we are proposing. To this end, we performed a set of simulation in which we suppose that the traffic splits, $r_{i,j}^k$, declared by users are affected by a relative error, so that the *actual traffic* split $\hat{r}_{i,j}^k$ is different from the declared one. We selected the same network topology as before, then generated border BRR configurations considering the declared $r_{i,j}^k$, and finally ran a set of simulations in which the traffic split was affected by a percentage of error ϵ, i.e.,

$$\hat{r}_{i,j}^k = (1 + \epsilon U) r_{i,j}^k,$$

where U is a random variable with uniform distribution between -1 and 1.

Fig. 7 reports the average, minimum and maximum overbooking probability experienced by BRRs for different values of ϵ. A Poisson traffic model is considered, with $N = 1$ (which is the worst-case). As can be noticed, the impact of ϵ is negligible on the average bandwidth availability experienced by users (being the error uniformly distributed); it has little effect on the minimum overbooking probability, while it increases the maximum values of the overbooking probability experienced by less lucky BRRs. Notice that for values of ϵ up to 50%, the maximum overbooking probability is still below the requested limit, and only 0.03% points larger than the limit values when the error is up to 100%. A natural conclusion is that the impact of errors in the traffic split marginally affects the proposed methodology. Even better results can be obtained when more complex traffic scenarios are considered (e.g. $N > 1$), as the worst-case analysis performed by the BRR admission criterion leaves more room to errors on the input parameters (see Fig. 4).

7.5. Routing optimization

After evaluating the effectiveness and robustness of the BRR admission criteria, we now focus our attention on the routing optimization algorithm. As a performance metric, we initially consider the maximum percentage of network capacity that can be allocated to guaranteed all BRRs, as captured by the statistical gain η. The impact of the path selection algorithm can be inferred from *border* configurations where, at each step of the B_i^k allocation, a new tree T_s is also generated according to the proposed heuristic. Therefore, more complex tree selection algorithms yield a better utilization of the network capacity.

1. *Impact of path set size and number of iterations:* To evaluate the impact of the two tunable parameters K (number of paths tested by the heuristic) and Z (number of iterations) introduced in Section 6.1, we report η versus the total network capacity. The impact of K and Z is of interest since it affects the complexity of the heuristic. Table 1 reports different border points versus different combinations of the Z and K parameters. Three different scenarios are considered, in which $d = 31,16,8$. The cases $Z = 0$ (or $K = 1$) are all equivalent, because in these cases the algorithm will immediately exit after the initial solution to the Steiner Tree problem is obtained (either no iterations are allowed, or no alternate paths can be considered to build other trees). Therefore, they can be considered as baseline configurations.

Table 1
η versus Z, K considering $d = 31, 16, 8$ destinations (top, medium, bottom table, respectively)

		K			
		1	2	3	4
$d = 31$					
Z	0	30.72	–	–	–
	1	–	32.69	33.13	33.20
	2	–	34.07	34.79	34.87
	3	–	35.50	36.01	36.06
SCTF		30.72			
$d = 16$					
Z	0	28.11	–	–	–
	1	–	34.84	34.98	35.63
	2	–	36.92	37.69	38.11
	3	–	38.29	39.49	40.01
SCTF		29.98			
$d = 8$					
Z	0	29.03	–	–	–
	1	–	34.04	34.98	35.35
	2	–	35.87	36.97	37.39
	3	–	36.05	37.32	36.67
SCTF		29.58			

For the sake of completeness, we report results obtained when considering the Steiner tree obtained with the so called Selective Closest Terminal First (SCTF) algorithm [10], which was shown to offer a good approximation of the optimal Steiner tree in generic meshed graphs.
Considering the impact of the number of paths K, a limited increase in η is observed, while the impact of the number of iterations, Z is larger. For example, considering $d = 31$, the increase of η goes from 30.72% up to 36.06% when $Z = 3$, $K = 4$, corresponding to an increase of 17%. This is largely due to the increased number of iterations Z rather than to the increased number of paths K. As an explanation, consider that, for each increase of Z by a unit, the optimization algorithm explores a different possible solution obtained by switching one path to a different destination. On the contrary, increasing the number K of paths generates solutions where a larger number of paths are tested from each destination; however, those paths are less desirable metrics (they are longer, in our case) and therefore waste more bandwidth when used to route traffic. This holds true for all scenarios, where the gains in the η increase up to 34% ($d = 16$) and to 28% ($d = 8$). The increased gains are due to a higher degree of variability of scenarios, allowing the optimization algorithm to obtain larger gains. Notice that the SCTF algorithm performs very similarly to the case when $K = 1$. This confirms the intuition that greedy algorithms do not provide good solutions, and that the local search criterion adopted by the proposed algorithm helps in improving the initial solution.

2. *Maximum number of iterations Z:* The previous results suggest that having just $K = 2$ paths for each pair of source/destination nodes guarantees an increase in total assured bandwidth. It is however interesting to study how large Z can grow before these gains become negligible. Fig. 8 gives the answer to the previous question, by reporting η versus Z for the three previously considered scenarios. As can be observed, all the curves in the plot show an asymptote: for $Z \to \infty$ the η is upper-bounded. In particular, for the $d = 31$ scenario, $Z \geqslant 8$ offers no negligible increase in the total assured bandwidth that can be successfully allocated to the AF traffic. For $d = 16$, $Z = 6$ it is sufficient to reach the maximum gain, while for $d = 8$, after $Z = 2$ iterations there is but a negligible gain.

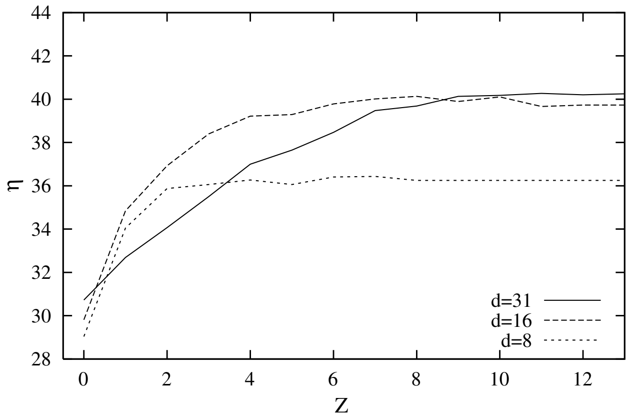


Fig. 8. Total assured bandwidth versus Z, considering $K = 2$ and $d = 31,16,8$.

To give the reader the intuition on the different resource allocation obtained for different values of K and Z, Fig. 9 plots the estimated overbooking probability experienced by BRRs in border configurations. Three BRRs are requested from each node, each BRR considers $d = 31$ destinations and requires an overbooking probability no larger than 0.1. BRRs are sorted in decreasing order of estimated overbooking probability when considering the case $Z = 0$, $K = 2$. Looking at the case in which $Z = 0$, the overbooking probability is in general smaller or equal that the limiting one, with few BRRs facing an overbooking probability exactly equal to the limit, as already notice in Fig. 3. Nonetheless, no increase in the allocated bandwidth can be performed without violating any BRR request. Considering instead increasing values of Z, we notice that, while the estimated overbooking probability is still within the QoS requirement, BRRs face larger overbooking probability. This corresponds to a better resource allocation. As already noticed in Fig. 8, the larger gains are obtained for small values of Z, while little difference can be appreciated among larger values of Z.

7.6. Dynamic scenario – blocking probability

While the total assured bandwidth is a performance index that is related to the maximum traffic a network can transport under bandwidth constraints, another important performance index is the Blocking Probability experienced by users when requesting a service. Smarter allocation algorithms allow network operators to accept a larger number of requests, and to increase the revenues. In this subsection, we explore how the proposed algorithm affects the BRR admission control in term of blocking probability. A dynamic scenario is considered, in which users issue BRRs to the network operator for a limited amount of time. A request is accepted in the network according to the admission policy, otherwise it is refused. BRRs arrive following a Poisson process of parameter λ, and the BRR holding time is exponentially distributed with average duration normalized to 1. Each BRR calls for an assured bandwidth $B_s = 1$ Mbit/s on links with a bandwidth B_l toward $d = 31$ egress nodes. The same network topology as in the previous section is considered.

Fig. 10 plots the blocking probability versus the offered load to the network, which is defined as $\rho = \frac{B_s}{B_l}\lambda$. The plot shows that a noticeable reduction of the blocking probability is already obtained for $Z = 2$, and a further decrease is achieved for

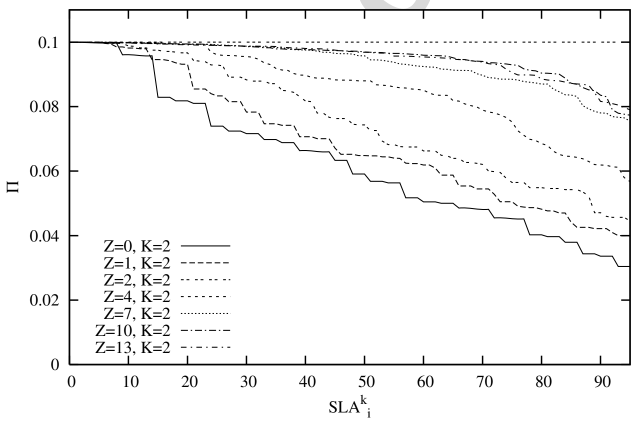


Fig. 9. Estimated overbooking probability of border configuration reached with different routing optimization parameters.

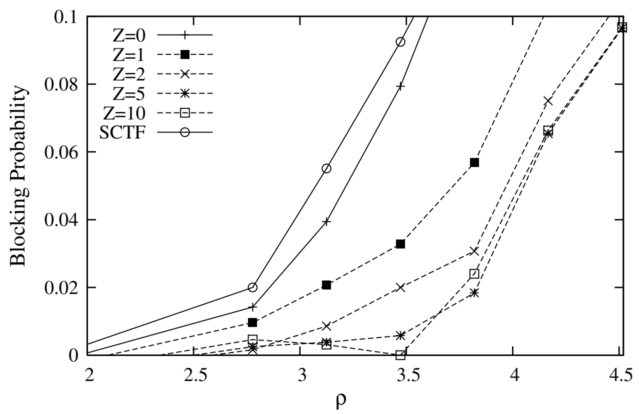


Fig. 10. Blocking probability of BRRs considering $K = 2$ and $d = 31$ for different values of Z.

$Z = 5$, while considering $Z = 10$ iterations yields little additional decrease.

This confirms the intuition that it is possible to better allocate network resources with limited complexity without violating the bandwidth constraints.

8. Conclusions

The paper proposed a low-complexity, heuristic algorithm for resource selection, allowing a network provider to increase the amount of bandwidth it can sell to its users without violating traffic guarantees. We introduced an admission control criterion which relies on the knowledge of the assured bandwidth and the set of involved destination (source) nodes. By assuming that BRRs are independent, we obtained a closed set of equations defining the amount of traffic that can be accepted by the network provider without violating requirement by BRRs. We then devised an optimization algorithm to define the set of links that will be used to route traffic belonging to each BRR, which is a generalized Steiner-tree problem. We proposed a simple, yet effective local search algorithm and evaluated its performance by considering several scenarios. Results show that, although the solution is non-optimal, it yields remarkable gains after a few iterations.

The BRR admission criterion and the routing optimization algorithm were proved to guarantee large network utilization while at the same time guaranteeing the bandwidth requested by already admitted traffic.

References

[1] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, W. Weiss, An architecture for differentiated services, RFC 2475, December 1998.

[2] V. Jacobson, K. Nichols, K. Poduri, An expedited forwarding PHB, RFC 2598, June 1999.

[3] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, Assured forwarding PHB group, RFC 2597, June 1999.

[4] M. Mellia, C. Casetti, G.Mardente, M. Ajmone Marsan, An analytical framework for SLA admission control in a DiffServ domain, IEEE Globecom 2002, Taipei, TW, November 2002.

[5] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol label switching architecture, RFC 3031, January 2001.

[6] D. Ooms, B. Sales,W. Livens, A. Acharya, F. Griffoul, F. Ansari, Overview of IP multicast in a Multi-Protocol Label Switching (MPLS) environment, RFC 3353, August 2002.

[7] Harry G. Perros, Khaled M. Elsayed, Call admission control schemes: a review, IEEE Communications Magazine 34 (November) (1996) 82–91.

[8] P. Winter, Steiner problem in networks: a survey, Networks 17 (Summer) (1987) 129–167.

[9] ANCLES – A Network Call-Level Simulator. URL: http://www.telematics.polito.it/ancles.

[10] S. Ramanathan, Multicast tree generation in networks with asymmetric links, IEEE/ACM Transactions on Networking 4 (4) (1996) 558–568.

[11] GT-ITM: Georgia Tech Internetwork Topology Models. http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html.

[12] N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, J.E. van der Merive, A flexible model for resource management in virtual private networks, in: Conference on Applications, technologies, architectures, and protocols for computer communication, Cambridge, MA, August 2003, pp. 95–108.

[13] L. Breslau, E.W. Knightly, S. Shenker, I. Stoica, H. Zhang, Endpoint admission control: architectural issues and performance, ACM Sigcomm, Stockholm, Sweden, August September 2001, pp. 57–69.

[14] X. Wang, H. Schulzrinne, Pricing network resources for adaptive applications in a differentiated services network, in: Proceeding of INFOCOM'2001, Anchorage, Alaska, pp. 943–952, April 2001.

[15] T.C.K. Hui, C.K. Tham, Adaptive provisioning of differentiated services networks based on reinforcement learning, IEEE Transactions on Systems, Man and Cybernetics 33 (4) (2003) 492–501.

[16] C. Oottamakorn, D. Bushmitch, A DiffServ measurement-based admission control utilizing effective envelopes and service curves, IEEE ICC 4 (June) (2001) 1187–1195.

Marco Ajmone Marsan is a Full Professor at the Electronics Department of Politecnico di Torino, in Italy. He is the Director of the Institute for Electronics, Information and Telecommunications Engineering of the National Research Council since September 2002. He is the Vice-Rector for Research, Innovation and Technology Transfer at Politecnico di Torino since November 2005. He has degrees in Electronic Engineering from Politecnico di Torino and University of California, Los Angeles.

He was at Politecnico di Torino Electronic Department from November 1975 to October 1987 – first as a researcher and then as an Associate Professor. He was a Full Professor at the University of Milan Computer Science Department from November 1987 to October 1990. During the summers of 1980 and 1981, he was with the Research in Distributed Processing Group, Computer Science Department, UCLA. During the summer of 1998 he was an Erskine Fellow at the Computer Science Department of the University of Canterbury in New Zealand.

In 1982, he received the best paper award at the Third International Conference on Distributed Computing Systems in Miami, Florida. In 2002, he was awarded a Honoris Causa Degree in Telecommunications Networks from the Budapest University of Technology and Economics. He is a corresponding member of the Academy of Sciences of Torino. He participates in a number of editorial boards of international journals, including the IEEE/ACM Transactions on Networking. He is listed by ISI among the highly cited researchers in Computer Science. He is a fellow member of IEEE.

Claudio Casetti graduated in Electrical Engineering from Politecnico di Torino in 1992 and received his Ph.D. in Electronic Engineering from the same institution in 1997. In 1995, he was a visiting scholar with the Networks Group of the University of Massachusetts, Amherst. In 2000, he was a visiting scholar with the Networking Group at UCLA. He is an Assistant Professor at the Dipartimento di Elettronica e Telecomunicazioni of Politecnico di Torino. He has coauthored about 80 journal and conference papers in the fields of networking and holds three patents. His interests focus on performance evaluation of TCP/IP networks and wireless communications. He is a member of IEEE.

Gianluca Mardente was born in Catanzaro, Italy, in 1976. He received his degree in Electronic Engineering in 2001, and a Ph.D in Telecommunications Engineering in 2005, both from Politecnico di Torino (Italy). His research interests include routing algorithms and QoS in IP networks, and traffic measurement, modelling and characterization. He currently works at Cisco Systems San Jose, CA, and is involved in the design of system performance analysis of next generation routers.

Marco Mellia was born in Torino, Italy, in 1971. He received his degree in Electronic Engineering in 1997, and a Ph.D in Telecommunications Engineering in 2001, both from Politecnico di Torino. From March to October 1999 he was with the CS department at Carnegie Mellon University as visiting scholar. Since April 2001, he is with Electronics Department of Politecnico di Torino as Assistant Professor. He participated in the program committees of several conferences including IEEE Infocom, IEEE Globecom and IEEE ICC. His research interests are in the fields of All-Optical Networks, Traffic measurement and modeling, QoS Routing algorithms. He is a member of IEEE.