Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Networks 50 (2006) 966–981
Computer Networks
www.elsevier.com/locate/comnet

Dynamic online QoS routing schemes: Performance and bounds

Antonio Capone [a,*], Luigi Fratta [a], Fabio Martignon [b]

[a] *Department of Electronics and Information (DEI), Politecnico di Milano, Piazza Leonardo da Vinci 32, Room 330, Third Floor, 20133 Milan, Italy*
[b] *Department of Management and Information Technology, University of Bergamo, 24044 Dalmine, Italy*

Received 14 August 2004; received in revised form 6 May 2005; accepted 20 June 2005
Available online 8 August 2005

Responsible Editor: H.J. Stuettgen

Abstract

Several dynamic QoS routing techniques have been recently proposed for new IP networks based on label forwarding. However, no extensive performance evaluation and comparison is available in the literature.

In this paper, after a short review of the major dynamic QoS routing schemes, we analyze and compare their performance referring to several networks scenarios. In order to set an absolute evaluation of the performance quality we have obtained the ideal performance of any routing scheme using a novel and flexible mathematical programming model that assumes the knowledge of arrival times and duration of the connections offered to the network.

This model is based on an extension of the maximum multi-commodity flow problem. Being an integer linear programming model, its complexity is quite high and its evaluation is constrained to networks of limited size. To overcome the computational complexity we have defined an approximate model, based on the multi-class Erlang formula and the minimum multi-commodity cut problem, that provides an upper bound to the routing scheme performance.

The performance presented in the paper, evaluated by measuring the connection rejection probability, shows that the schemes considered reach, in several scenarios, the ideal performance, showing that no much gain is left for alternate new schemes.

Keywords: QoS Routing; Mathematical models; MPLS; Flow deviation

* Corresponding author. Tel.: +39 02 2399 3449; fax: +39 02 2399 3413.
E-mail addresses: capone@elet.polimi.it (A. Capone), fratta@elet.polimi.it (L. Fratta), fabio.martignon@unibg.it (F. Martignon).

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2005.06.007

1. Introduction

The current evolution of Internet architecture is towards service differentiation and Quality of Services (QoS) support [1]. In order to offer guaranteed end-to-end performance (as bounded delay, jitter or loss rate), it is necessary to introduce some sort of resource reservation mechanism and traffic control. With classical IP routing, however, when the resources are not available on the shortest path, the connection request is rejected even if sufficient resources exist on alternative paths.

With new label based forwarding mechanisms, such as Multi Protocol Label Switching (MPLS) [2] and Generalized MPLS (GMPLS) [3,4], per flow path selection is possible and QoS parameters can be taken into account by routing algorithms. The goal of QoS routing schemes is to select a path for each traffic flow (micro-flows or aggregated-flows according to routing granularity) that satisfies quality constraints based on the actual available resources in the network.

The QoS requirement of a connection can be given as a set of constraints on links and paths. For instance, bandwidth constraints require that each link on the path has sufficient bandwidth to accommodate the connection.

From the user point of view QoS routing algorithms must satisfy the QoS requirements, while from the provider point of view they have also to maximize the resource utilization.

The QoS routing algorithms proposed in the literature [5–11] can be classified into static or dynamic, and online (on demand) or offline (precomputed) [6]. Static algorithms use only network information that does not change in time, while dynamic algorithms use the current state of the network, such as available link capacity. In online routing algorithms, path requests are considered one by one, and usually previously routed connections cannot be rerouted. Offline routing does not allow new path route computation and it is usually adopted for permanent connections.

This paper is focused on the performance evaluation of dynamic online QoS routing algorithms, where the maximum resource utilization is achieved by minimizing connection rejection probability of future requests.

First, we review some of the most popular algorithms proposed in the literature, such as the Min-Hop Algorithm (MHA) [12], the Widest Shortest Path Algorithm (WSP) [13], the Minimum Interference Routing Algorithm (MIRA) [14,15], the Profile-Based Routing algorithm (PBR) [11] and the Virtual Flow Deviation (VFD) algorithm [16]. We describe in some detail MIRA, PBR and VFD algorithms. These algorithms take explicitly into account the topological layout of the ingress and egress points of the network. The VFD algorithm, recently proposed in [16], considers also the traffic statistics. More precisely, VFD exploits the knowledge of the layout of the ingress/egress nodes of the network, and uses the statistics information about the traffic offered to the network in order to forecast future connections arrivals.

Then, to provide a measure of the quality of the performance, we present some theoretical bounds to the performance achievable by any online QoS routing algorithm by means of two novel and flexible mathematical models.

The first one, Ideal Routing (IR), is an Integer Linear Programming model and is based on an extension of the maximum multi-commodity flow problem [17]. It provides an optimal routing configuration capable of accommodating the traffic offered to the network. The model minimizes the number of rejected connections assuming that the connection arrival times and their durations are known. Accepted connections are provided a single path which is maintained for the whole connection lifetime (no re-routing is allowed). The IR model describes an ideal routing scheme that achieves the minimum connection rejection probability. However, due to the complexity of its formulation, the solution of this model requires long computing time and large memory, even with state of the art optimization tools [18,19]. Therefore, its applicability is limited to small size network scenarios.

The second model, based on the multi-class Erlang formula and on the minimum multi-commodity cut problem [20–22] (Min-Cut model), is an approximate one and provides a looser lower bound to the connection rejection probability. It can be applied to larger and more complex network topologies since its memory occupation and computing time are considerably lower than in the first model.

The numerical results on the performance of the algorithms considered have been obtained by simulating a set of relevant network scenarios. The comparison of these results with the bounds obtained with the IR and Min-Cut models shows that the VFD algorithm performs quite close to the ideal algorithm.

The paper is structured as follows: in Section 2 we address the QoS routing problem and we review some existing routing algorithms. In Section 3 we review the Virtual Flow Deviation algorithm, pointing out its innovating features. In Section 4 we illustrate the IR model, discussing the problem of setting the objective function parameters, and the Min-Cut model. In Section 5 we analyze and discuss the performance of online algorithms under a variety of network scenarios, comparing their performance to the theoretical bounds calculated using the mathematical models. Section 6 concludes the paper.

2. Dynamic online QoS routing schemes

In this section, we review some of the most relevant dynamic QoS algorithms proposed in the literature. In the following we assume that all the quality parameters requested by incoming connections can be controlled by defining an equivalent flow bandwidth as discussed in [23,24]. This assumption allows us to focus only on bandwidth constraints.

Let a network be represented by a graph $G(N,A)$, where the nodes N represent routers and arcs A represent communication links, as shown in Fig. 1.

The traffic enters the network at ingress nodes S_i and exits at egress nodes T_i. Each connection requires a path from S_i to T_i. The capacity C_{ij} and the actual flow F_{ij} are associated to each link (i,j), where F_{ij} is defined as the total amount of bandwidth allocated to the connections that pass through link (i,j). The residual bandwidth of link (i, j) is defined as $R_{ij} = C_{ij} - F_{ij}$.

A new connection can be routed only over links with R_{ij} greater or equal to the requested bandwidth. Referring to a new connection k with requested bandwidth d_k, a link is defined as *feasible* if $R_{ij} \geqslant d_k$. The feasible network for connection k is the sub-graph of G obtained by removing all un-feasible links. A connection can be accepted if at least one path between S_i and T_i exists in the feasible network. The minimum R_{ij} over a path defines the residual bandwidth of that path.

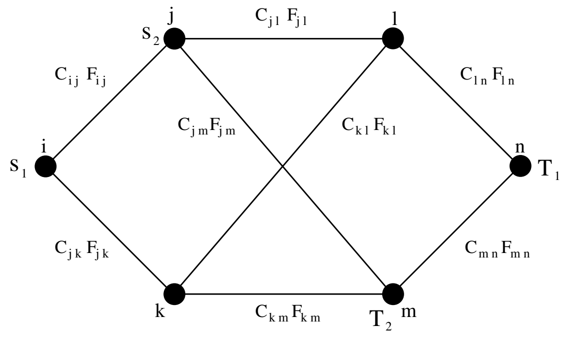

Fig. 1. QoS network state.

2.1. Min-hop algorithm

The Min-hop algorithm (MHA) [12] routes a new connection along the path, between source and destination, with the minimum number of feasible links.

This scheme, based on the Dijkstra algorithm, is simple and computationally efficient. However, being the cost given to each link independent of the current link load, MHA tends to use the same paths until saturation is reached before switching to other paths with less utilized links. This can result in an unbalanced routing with heavily loaded bottlenecks.

2.2. Widest shortest path algorithm

The Widest shortest path algorithm (WSP) [13] is an improvement of the Min-Hop algorithm, as it attempts to balance the network load. In fact, WSP chooses a feasible path with minimum number of links and, if there are multiple such paths, the one with the largest residual bandwidth, thus discouraging the use of already heavily loaded links.

However, WSP still has the same drawbacks as MHA since the path selection is performed among the shortest feasible paths that are used until saturation before switching to longer feasible paths.

2.3. Minimum interference routing algorithm

The Minimum Interference Routing Algorithm (MIRA) [14,15] explicitly takes into account the location of the ingress and egress routers. The key idea of MIRA is to route a new connection over a path which least interferes with possible future requests.

More specifically, the connection request between (S_i, T_i) is routed aiming to maximize an objective function which is either the minimum maximum-flow (max-flow) of all other ingress/egress pairs, defined as the maximum amount of flow that can be transferred from the ingress to the corresponding egress node [17], or a weighted sum of max-flows, where weights α_{ST} assigned to each ST pair reflect the "importance" of the flow.

In order to achieve an online routing algorithm, MIRA keeps an updated list of the critical links, i.e. the links whose use by the new call reduces the max-flow between other pairs.

When a new call has to be routed between the source/destination pair (S_i, T_i), MIRA determines the set L_{ST} of the critical links for all the source/destination pairs (S_j, T_j) with $j \neq i$. For each link l, a weight $w(l)$ is defined as $w(l) = \sum_{(S,T):l \in L_{\mathrm{ST}}} \alpha_{\mathrm{ST}}$. The new call is then routed on the shortest path considering the lengths $w(l)$, $l \in A$, assigned to the links.

In spite of its more sophisticated functions, MIRA still has the following limitations whose effect will be shown when discussing the numerical results:

- MIRA discourages the use of critical links based only on the number of other S–T pairs that could use them, without verifying if these S–T pairs actually use these links. Furthermore, it does not consider the amount of traffic generated by S–T pairs. As a consequence, MIRA preserves the use of some links that remain under-utilized, thus causing a suboptimal use of the network. To overcome this limitation, it has been proposed to maximize a weighted sum of the source/destination max-flows. However, in [14,15] the weights are chosen offline and do not adapt to changes in network traffic. Hence this solution does not provide the flexibility expected in an online routing scheme.
- MIRA sets link weights almost in a static way. As a result, the only event that can cause the redistribution of new weights is the saturation of some links, similarly to the MHA.
- While choosing a path for a new request, MIRA does not take into account how this connection will affect the future requests of the same ingress/egress pair.

2.4. Profile-based routing

The Profile-Based Routing algorithm (PBR) [11] exploits, like MIRA, the topological information about ingress/egress nodes of the network. Moreover, it takes into account network traffic statistics by estimating network traffic profiles, obtained by measurements of service-level agreements established with network users, as a prediction of future traffic distribution.

PBR is based on an offline preprocessing step that determines the amount of bandwidth allocated to each traffic class on network links. Based on this allocation PBR performs an admission control on incoming connections. This feature considerably reduces the complexity of the computation performed online upon a new connection request. However its performance is limited since the admission control might reject a new call event if a feasible path in the network actually exists.

In the following the performance of PBR is not considered since we focus our analysis on pure online routing algorithms that do not reject incoming connections if a feasible path exists. However, the mathematical models we propose in this paper can be easily adapted to evaluate bounds on the performance of the PBR algorithm.

3. Virtual flow deviation

The Virtual Flow Deviation (VFD) is a new routing algorithm, recently presented by the authors [16], that aims to overcome the limitations of the routing algorithms just reviewed by exploiting all the information available when a route selection must be taken. This algorithm is described in some more details since it is not yet well known in the research community.

To better describe the current state of the network and to forecast its future state, VFD exploits the topological information on the location of ingress/egress pairs, used by MIRA, as well as the traffic statistics obtained by measuring the load offered to the network at each source node. This information plays a key role in choosing the best route of a new request in order to prevent network congestion.

To account for the future traffic offered to the network, VFD routes not only the real call, but also some virtual calls which represent an estimate (based on measured traffic statistics) of the connection requests that are likely to interfere with the current real call. The number of these virtual calls, as well as the origin, destination, and the bandwidth requested should reflect as closely as possible the real future conditions of the network. These parameters can be estimated based on the past traffic statistics of the various ingress/egress pairs, as detailed in Section 3.1.

The accuracy of the measured traffic statistics is an important factor for the performance of the algorithm, since very inaccurate traffic statistics can severely affect the performance of VFD. However, the traffic statistics that are usually obtained with real-time traffic measurements in IP networks are sufficiently accurate to provide better performance than simply using the topological information as performed by MIRA.

All the information on the network topology and the estimated offered load is used to select a path which uses at best the network resources and minimizes the number of rejected calls. Such a path selection is performed in VFD by the Flow Deviation method [25,26], which allows to determine the optimal routing for all connections entering the network.

3.1. The virtual calls

To implement VFD we have to determine how many virtual calls should be generated, their source/destination pairs, and their bandwidth requests. In this process, we can easily measure and assign to each S–T pair the two following parameters:

- the average traffic (λ_{S_i,T_i}) offered by the S–T pair i, defined as the average number of connections entering the network through the node S_i with destination T_i in the interval Δt;
- the probability distribution of the bandwidth requested at each S–T pair, estimated as the ratio between the number n_d of calls which have requested d bandwidth units and the total number N of calls considered for the estimation.

Note that, for sake of simplicity, we have considered bandwidth requests that are integer multiples of the bandwidth unit. However, the algorithm works also with real values bandwidth requirements.

If the total average load offered to the network, Λ, is defined as:

$$\Lambda = \sum_{\forall \text{ pairs } S_i - T_i} \lambda_{S_i,T_i},$$

the probability P_{S_i,T_i} to receive a call between the node pair (S_i, T_i) is given by $P_{S_i,T_i} = \lambda_{S_i,T_i}/\Lambda$, while the probability to have a request of d bandwidth units at the source node i for the destination node T_i, P_{d_i}, is estimated by $P_{d_i} = n_d/N$.

The parameters (S_i, T_i, d_i), which completely determine the virtual calls, are random variables generated according to the probability density functions P_{S_i,T_i} and P_{d_i}.

To determine the number N_v of virtual calls to be generated, two different approaches have been proposed in [16]. In this paper we consider the approach that evaluates the number of virtual calls as $N_v = \lfloor (N_{\max} - N_{\mathrm{A}}) \rfloor$, where $N_{\max}$ is an estimate of the maximum number of calls that can be routed in the network, and N_{A} is the number of already active calls. In our simulation, $N_{\max}$ is computed at the beginning by loading the empty network by randomly generating connections according to traffic statistics until network is saturated. In real network operations, $N_{\max}$ should be recomputed when traffic statistics change significantly. The virtual calls are routed together with the real call, represented by (S_R, T_R, d_R), using the Flow Deviation method.

3.2. The virtual flow deviation algorithm

The VFD algorithm operation is described in the flow diagram of Fig. 2.

Upon a new call request, N_v virtual calls are generated. The real call and the virtual calls are then offered to the network. The procedure to route the new traffic operates in two steps.

In the first step an initial feasible flow assignment is obtained. Calls are routed one by one starting from the real call. A call can be either defined as ACTIVE, if a feasible path has been found, or NON ACTIVE otherwise. The procedure stops if the real call is declared NON ACTIVE, otherwise it is repeated until all calls have been considered.

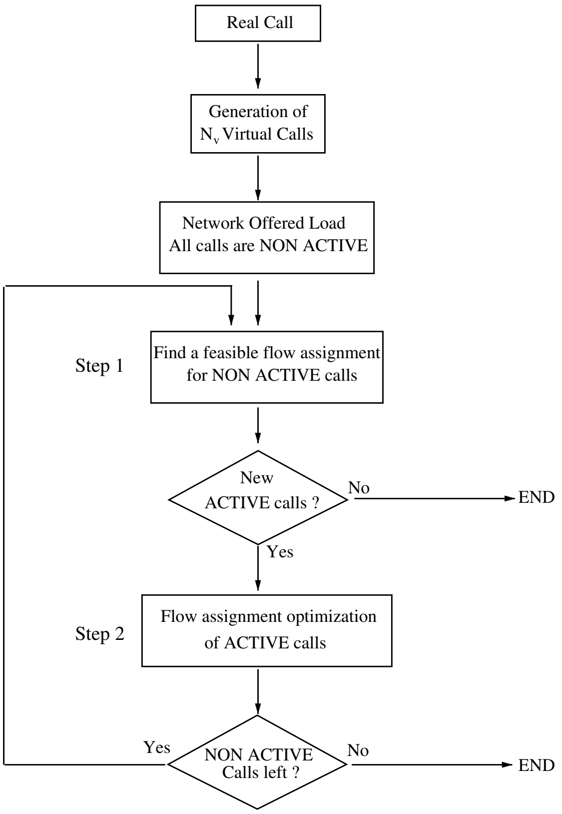

Fig. 2. The virtual flow deviation algorithm.

In step two the routing of all ACTIVE calls is optimized using the Flow Deviation Method [25,26].

Then step one is repeated for the NON ACTIVE calls. If at least one NON ACTIVE call is declared ACTIVE, step two is repeated and the procedure is iterated until either all calls are ACTIVE or step one does not define any new call as ACTIVE.

The feasible flow assignment is obtained in step one by using the Shortest Path Algorithm (Dijkstra) applied to the network whose links weights reflect the actual channel utilization. More specifically, for each link a weight $w_{ij} = \frac{1}{C_{ij} - F_{ij}}$, is assigned and updated at each iteration.

A more formal description of VFD is given by the pseudo-code in Table 1.

According to the above described procedure the new real call is routed on the optimum path considering the expected future evolution of the network traffic load.

3.3. Complexity comparison

Table 2 summarizes the information needed by the considered algorithms. The Min-Hop (MHA)

Table 1
Pseudo-code specification of *Step 1* and *Step 2* introduced in Fig. 2

```
for (∀ connection (S_k, T_k, d_k, flag_k))
  flag_k = NON ACTIVE
end for
do
  for (∀ connection (S_k, T_k, d_k, flag_k) = NON ACTIVE)
    for (∀ link l_ij)
      weight assignment:
      w_ij = 1/(C_ij − F_ij) if F_ij < C_ij
      w_ij = ∞ if F_ij = C_ij
    end for
    execution of Dijkstra shortest Path algorithm:
    if (∃ a path between S_k and T_k with bandwidth d_k)
        update F_ij and memorize the path
        flag_k = ACTIVE
    end if
  end for
  for (∀ connection (S_k, T_k, d_k, flag_k) = ACTIVE))
    execution of the Flow Deviation method
  end for
While (in the last iteration at least one flag_k has been
set to ACTIVE)
```

Table 2
Comparison of the information needed by the Min-Hop, Widest-Shortest Path, MIRA, PBR and VFD algorithms

	MHA	WSP	MIRA	PBR	VFD
Network topology	X	X	X	X	X
Residual capacity (R_{ij})	X	X	X	X	X
Location of ingress/ egress nodes			X	X	X
Traffic statistics				X	X

and the Widest-Shortest Path (WSP) algorithms use only topological information and residual bandwidth of network's links, while VFD, MIRA and PBR exploit the knowledge of the layout of the ingress/egress nodes of the network. This quasi-static information can be easily collected with little extra effort [15].

In addition, VFD and PBR use the statistics information about the traffic offered to the network in order to forecast future connections arrivals. This information can be obtained with slightly higher effort using traffic measurements already performed in IP networks.

4. Mathematical models

In this section, we introduce two novel mathematical models that provide bounds to the performance achievable by any dynamic online routing algorithm.

The first model, Ideal Routing (IR), assumes the exact knowledge of future traffic. The routing decisions are taken to optimize the operation of the network loaded with the actual present and future traffic. No practical routing scheme can perform better. The model is based on an extension of the maximum multi-commodity problem and its solution obtained by ILP.

The second model, (Min-Cut), first computes the maximum multi-commodity flow, that represents a lower bound to the capacity of the minimum multi-commodity cut (min-cut) of the network [21,22]. Then, the connection rejection probability for the given network scenario is obtained by using the multi-class Erlang formula, assuming that the min-cut capacity of the network can be fully exploited. The solution of this model is easier to obtain than IR but provides a looser lower bound to the rejection probability.

To the best of our knowledge, our ILP model is the first that takes explicitly into account the arrival times and duration of the connections offered to the network, and that models very closely the behavior of online QoS routing algorithms. Its formulation is quite flexible and with simple modifications it can model the behavior of offline routing algorithms and can take into account connections rerouting, split flows and link capacity changes.

Concerning the Min-Cut model, the new contribution is to combine the computation of the min-cut capacity with the use of the multi-class Erlang formula. This allows to compute a lower bound to

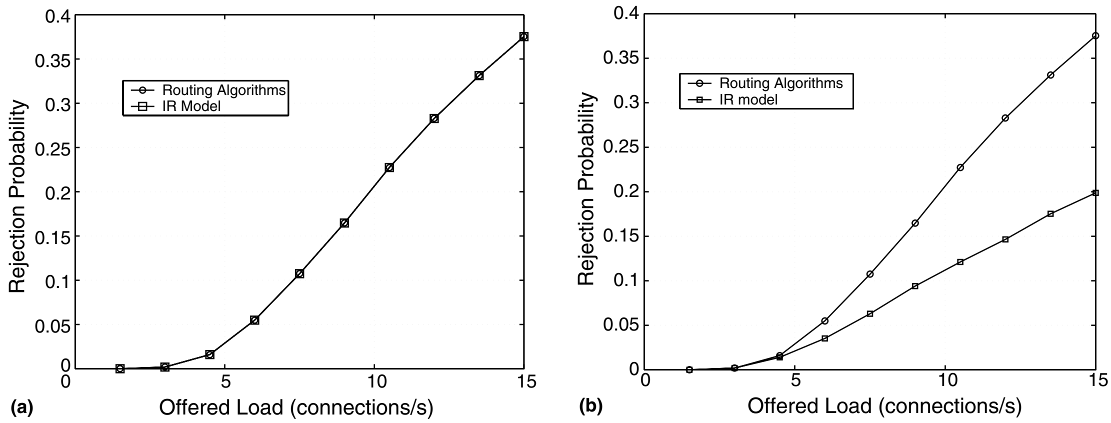

Fig. 3. Connection rejection probability versus the average total load offered to a single link with: (a) $b_k = 2^{N_c - k}$; (b) $b_k = 1$.

the connection rejection rate in more complex network topologies where the ILP model is too cumbersome.

4.1. Ideal routing model

The basic assumption of the IR model is the knowledge of future traffic offered to the network. Let $K = \{1,\ldots,N_c\}$ be the set of connections, each one represented by the triplet (S_k, T_k, d_k), that specify source node, destination node and requested bandwidth. Connection k is further characterized by its arrival time, t_k and its duration τ_k. Given N_c connections (Fig. 4 shows an example for $N_c = 4$), the time interval from the arrival of the first connection and the last ending time of a connection is subdivided in a set I of $2N_c - 1$ time intervals. In each time interval, t, the number of active connections $M(t)$ remains constant. This number changes by one from interval to interval: it increases if a new connection arrives, and decreases if a connection ends. Let us denote with $B(k)$ the time interval beginning at the arrival time of connection k, and with I_k the set of time intervals in which connection k is active. The function $M(t)$ corresponding to the N_c connections offered in Fig. 4 is shown in Fig. 5.

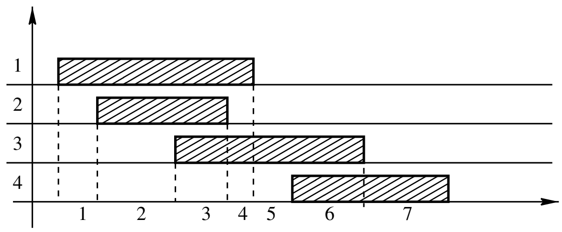

Fig. 4. Arrival time and duration of the connections offered to the network.

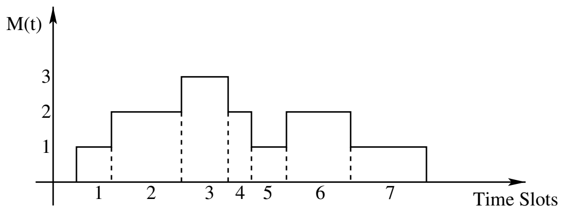

Fig. 5. Number $M(t)$ of active connections in each time slot.

Given the function $M(t)$, the optimum routing must minimize the call rejection probability. This optimization problem can be formulated as Integer Linear Programming (ILP) if the following notations and definitions are adopted.

Let $G = (N, A)$ be the direct graph representing the network. Let $n = |N|$ and $m = |A|$ be the number of nodes and arcs, respectively. The capacity C_{ij} is associated to each arc (i,j).

For each connection k, $k \in K$, create two new nodes SS_k and TT_k and two new directed arcs, (SS_k, S_k) and (T_k, TT_k), of infinite capacity. Let SN and TN be the sets of the added nodes containing all SS_k and TT_k, respectively. Similarly, let A_{SN} and A_{TN} be the sets of arcs containing all (SS_k, S_k) and (T_k, TT_k), respectively.

Finally let $G' = (N', A')$ with $N' = N \cup SN \cup TN$ and $A' = A \cup A_{\mathrm{SN}} \cup A_{\mathrm{TN}}$.

Based on the above definitions and notation, we establish the ILP formulation of the IR model. To this purpose, let us define the following decision variables:

$$x^k_{ijt} = \begin{cases} 1 & \text{if connection } k \text{ is routed on arc}(i,j) \\ & \quad \text{in time slot } t, \\ 0 & \text{otherwise.} \end{cases}$$

for $(i,j) \in A'$, $k \in K$ and $t \in I$. We force $x^k_{ijt} = 0, \ \forall\, t \notin I_k$.

Since the goal is to minimize the connection rejection probability, we can equivalently maximize the number of connections accepted by the network. The problem can thus be formulated as follows:

$$\text{Maximize} \quad \sum_{k\in K} b_k \cdot x^k_{SS_kS_kB(k)}, \tag{1}$$

$$\text{s.t.} \quad \sum_{k\in K} d_k \cdot x^k_{ijt} \leqslant C_{ij} \quad \forall (i,j) \in A, \ t \in I, \tag{2}$$

$$\sum_{(j,l)\in A'} x^k_{jlt} - \sum_{(i,j)\in A'} x^k_{ijt} = \begin{cases} 1 & \text{if } j \in SN \\ 0 & \text{if } j \in N \\ -1 & \text{if } j \in TN \end{cases} \quad \forall k \in K, j \in N', \ t \in I \tag{3}$$

$$x^k_{ijt} = x^k_{ijB(k)} \quad \forall k \in K, (i,j) \in A',\ t \in I_k, \tag{4}$$

$$x^k_{ijt} \in \{0,1\} \quad \forall k \in K, (i,j) \in A',\ t \in I_k, \tag{5}$$

The objective function (1) is the weighted sum of the connections accepted in the network, where b_k represents the benefit associated with connection k. Different settings of b_k are possible, and they reflect different behaviors of the model as discussed later.

Constraints (2) ensure that, at each time slot, the total flow due to all the connections that use arc (i,j) does not exceed the arc capacity, C_{ij}, for all $(i,j) \in A$.

Constraints (3) represent the flow balance equations expressed for each node belonging to the extended graph G', in each time slot $t \in T$. Note that these constraints define a path for each connection between its source and destination nodes.

Constraints (4) impose that the accepted connections cannot be aborted or rerouted for their entire lifetime.

Finally, requiring that the decision variables in (5) are binary implies that each connection is routed on a single path.

The online QoS routing algorithms we are considering in this paper do not reject a new connection with (S_k, T_k, d_k) if at least one path with a residual available bandwidth greater than or equal to the requested bandwidth d_k exists.

To account for this feature, the objective function (1) must be properly set. To this purpose it is sufficient to set:

$$b_k = 2^{N_c - k}, \tag{6}$$

having numbered the N_c connections from 1 to N_c according to their arrival times. With such a setting of b_k the benefit to accept connection k is always greater than the benefit of accepting, instead, all the connections from $k+1$ to N_c, since:

$$2^{N_c - k} > \sum_{i=k+1}^{N_c} 2^{N_c - i}. \tag{7}$$

This choice of the weights b_k allows the mathematical formulation to model very closely the behavior of real online routing algorithms. To verify the accuracy of the model we have considered a simple scenario where a single link connects a source/destination pair. We have obtained the performance in the case of channel capacity equal to 20 bandwidth units and assuming the bandwidth b_k to be uniformly distributed between 1 and 3 units and the lifetime τ_k to be exponentially distributed with mean 15 s.

In this simple case all the routing algorithms provide the same performance since only one path exists between source and destination. The rejection probability shown in Fig. 3(a) has been computed using the multi-class Erlang Formula. The bound provided by the IR model completely overlaps the online routing performance. Note that different choices of b_k provide different IR Model performance. For instance, selecting $b_k = 1$ for all k we obtain the performance shown in Fig. 3(b). The large reduction in rejection probability is expected since the optimization of the objective function will result in rejecting connections with high bandwidth requirements and long lifetime in favor of smaller and shorter ones. The difference between the bound and the real performance increases as the network load increases.

The above stated ILP formulation is quite general and by simply modifying the constraints (2)–(5) allows to solve different problems. In the following a few alternative modeling formulations are outlined.

Removing constraints (4), the network can possibly change the path of a connection at each time slot.

If the constraints (5) are relaxed, the problem formulation is no longer an integer program and its solutions requires less computing time and memory occupation. However, in this formulation each connection can be split over multiple paths. Such splitting, that requires packet reordering, is not always tolerated by end user's applications that may use transport protocols like TCP.

Finally, it is possible to consider link capacities that change in time slot by slot. To include this feature, that allows to take into account link failures or variations in the available capacity along a path, it is sufficient to substitute the constant parameter C_{ij} with a time-varying one, C_{ijt}.

The above model has been implemented using the AMPL language [19], and solved using the

CPLEX solver [18]. This problem formulation, however, involves a large number of decision variables, $mN_c(2N_c - 1) \approx 2mN_c^2$, since a variable is associated to each arc (m), to each connection (N_c) and to each time interval ($2N_c - 1$). For practical size network, and for a large number of offered connections the memory occupation requested by CPLEX can be too large. Fortunately, the number of variables to be considered can be reduced without loosing optimality in the solution by observing that not all the $2N_c - 1$ time slots must be considered in the optimization process but only those that correspond to local maxima of $M(t)$.

In fact, the connection requests in non local maxima time slots are a subset of those active in the relevant local maxima. This condition implies that the corresponding requirements in such time slots are dominated by those corresponding to local maxima and therefore have no impact on the optimum solution.

The reduction in the number of decision variable that has been observed in the several examples considered has been remarkable. However, in the worst case, half of the time slots correspond to local maxima and the number of variables to be considered is bounded by mN_c^2.

4.2. Min-cut model

In this section, we propose a second mathematical model that allows to determine a lower bound to the connection rejection probability. The solution of this model has computing times and memory occupation considerably lower than the previous one. However, in some scenarios, the bound obtained can be quite lower than the value provided by IR.

Let us consider a directed graph $G = (N, A)$ defined by a set of nodes, N, and a set of arcs, A each one characterized by a capacity C_{ij}. A set of source/destination pairs $K = \{1, \ldots, N_s\}$, indicated by S_i and T_i, respectively, $i \in K$, is also assigned. Each source S_i generates a flow αf_i, towards destination T_i, that can be split over multiple paths. The problem is to find the maximum α, indicated by α^*, such that for all $i \in K$ the flow quantities $\alpha^* f_i$ can be routed to their destinations.

The solution to this problem, obtained via linear programming techniques, provides the maximum multi-commodity flow $F_{\max} = \sum_{i\in K} \alpha^* f_i$. Note that $F_{\max}$ represents a lower bound to the capacity of the minimum multi-commodity cut of the network, as discussed in [21,22].

Once $F_{\max}$ has been obtained, the connection rejection probability for the given network scenario is obtained by using the multi-class Erlang formula with $F_{\max}$ servers [20] that is briefly reviewed in the following.

Let us consider N different traffic classes offered to a network system with C servers. The connections belonging to the class i request d_i bandwidth units. The connections arrival process is a Poisson process with average λ_i, while the connections duration is distributed according to a generic distribution $f_{\Theta_i}(\theta_i)$. Let $\Lambda = \sum_{i=1}^{N} \lambda_i$ be the total load offered to the network.

An appropriate state description of this system is $n = (n_1, \ldots, n_N)$, where $n_i, i = 1, \ldots, N$ is the number of connections belonging to the class i that occupy the servers. The set of all the possible states Ω is expressed as $\Omega = \{n | X \leqslant C\}$, where X, the total occupation of all the servers, is given by $X = \sum_{i=1}^{N} n_i d_i$.

If we indicate with $A_i = \lambda_i\, E[\Theta_i]$ the traffic offered to the network by each class, the steady state probability of each state is simply given by the multi-class Erlang formula:

$$\pi(n) = \frac{1}{G} \prod_{i=1}^{N} \frac{A_i^{n_i}}{n_i!}, \tag{8}$$

where G is the normalization constant that ensures that the $\pi(n)$ sum to 1 and it has therefore the following expression:

$$G = \sum_{n\in\Omega} \pi(n) = \sum_{n\in\Omega} \left(\prod_{i=1}^{N} \frac{A_i^{n_i}}{n_i!} \right). \tag{9}$$

Using the steady state probability calculated with equation (8) we can derive the loss probability of the generic class i, Π_i, as follows:

$$\Pi_i = \sum_{n\in B_i} \pi(n), \tag{10}$$

where B_i is the set of the blocking states for the class i, defined as $B_i = \{n | C - d_i < X \leqslant C\}$. The overall connection rejection probability, p_{rej}, is then given by:

$$p_{\text{rej}} = \sum_{i=1}^{N} \frac{A_i \Pi_i}{\sum_{i=1}^{N} A_i}. \tag{11}$$

If we substitute C with the maximum multi-commodity flow value $F_{\max}$ in all the above expressions, we can compute the connection rejection probability using equation (11).

In network topologies with high link capacities, $F_{\max}$ can assume high values, and the enumeration of all the allowed states becomes computationally infeasible, since the cardinality of Ω is of the order of $F_{\max}^N$ [27]. In these network scenarios, Eqs. (8)–(11) are computationally too complex so we propose to apply the algorithm described in [27,28] that computes recursively the blocking probability based on the peculiar properties of the normalization constant G. For network topologies with very high link capacities we implemented the inversion algorithm proposed in [29] to compute the blocking probabilities for each class.

5. Numerical results

In this section, we compare the performance, measured by the percentage of rejected calls versus the average total load offered to the network, of the Virtual Flow Deviation algorithm, the Min-Hop Algorithm and MIRA with the bounds provided by the mathematical models presented in the previous section referring to different network scenarios in order to cover a wide range of possible environments.

The first scenario we consider is illustrated in Fig. 6. In this network the links are unidirectional with capacity equal to 120 bandwidth units. In the following capacities and flows are all given in bandwidth units. The network traffic, offered through the source nodes S_1, S_2 and S_3, is unbalanced since sources S_2 and S_3 generate a traffic four times larger than S_1. Each connection requires a bandwidth uniformly distributed between 1 and 3. The lifetime of the connections is assumed to be exponentially distributed with average equal to 15 s.

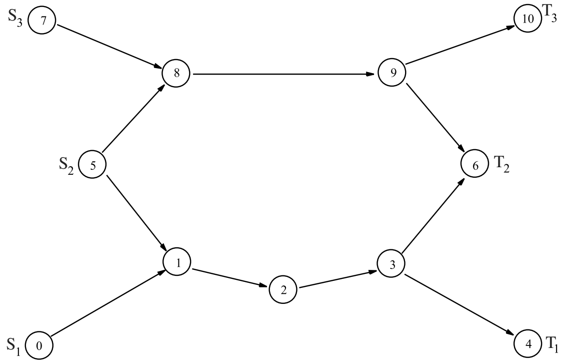

Fig. 6. Network topology with unbalanced offered load: the source/destination pairs S_2–T_2 and S_3–T_3 offer to the network a traffic load which is four times higher than that offered by the pair S_1–T_1.

In this simple topology connections S_1–T_1 and S_3–T_3 have one path only, while connections S_2–T_2 have two different paths.

The rejection probability versus the offered load for MIRA, MHA, VFD, IR and Min-Cut models are shown in Fig. 7. The poor performance of MIRA is due to its lack of considering any information about the load distribution in the network. In this particular topology, due to critical links (1,2), (2,3) and (8,9), S_2–T_2 connections are routed on the path (5–8–9–6) that contains the minimum number of critical links. MHA, that selects for connections S_2–T_2 the path with the minimum

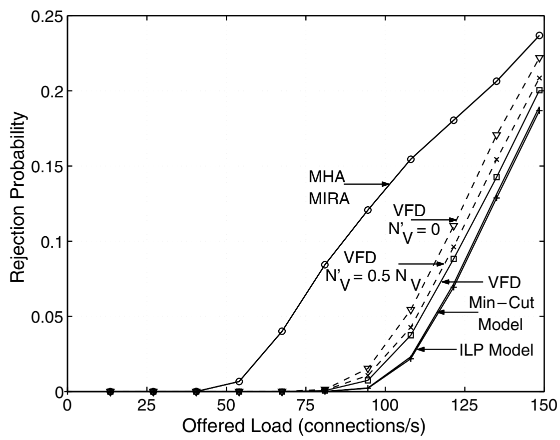

Fig. 7. Connection rejection probability versus the average total load offered to the network of Fig. 6.

number of hops, routes the traffic as MIRA and their performances overlap. Better performance is achieved by VFD. Since its behavior depends on the number of virtual connection N'_v used in the routing phase, we have considered three cases: $N'_v = 0$, $N'_v = 0.5 \cdot N_v$ and $N_v = \lfloor (N_{\max} - N_{\mathrm{A}}) \rfloor$. In the first case, even if no information on network traffic statistics is taken into account, the VFD algorithm achieves much better performance than previous schemes due to the better traffic balance provided by the Flow Deviation algorithm. Only when the offered load reaches very high values the improvement reduces. The third case corresponds to the VFD version described in Section 3.2 that takes most advantage from traffic information. The best performance has been measured and the gain over existing algorithms is provided even at high loads. An intermediate value of N'_v (case 2) provides, as expected, intermediate performance. As far as the performance of the two mathematical models, we observe that the approximate Min-Cut model curve overlaps that of the IR model. Note that VFD performs very close to the theoretical bounds in this scenario.

In the same network scenario we have verified that the VFD algorithm practically reaches the

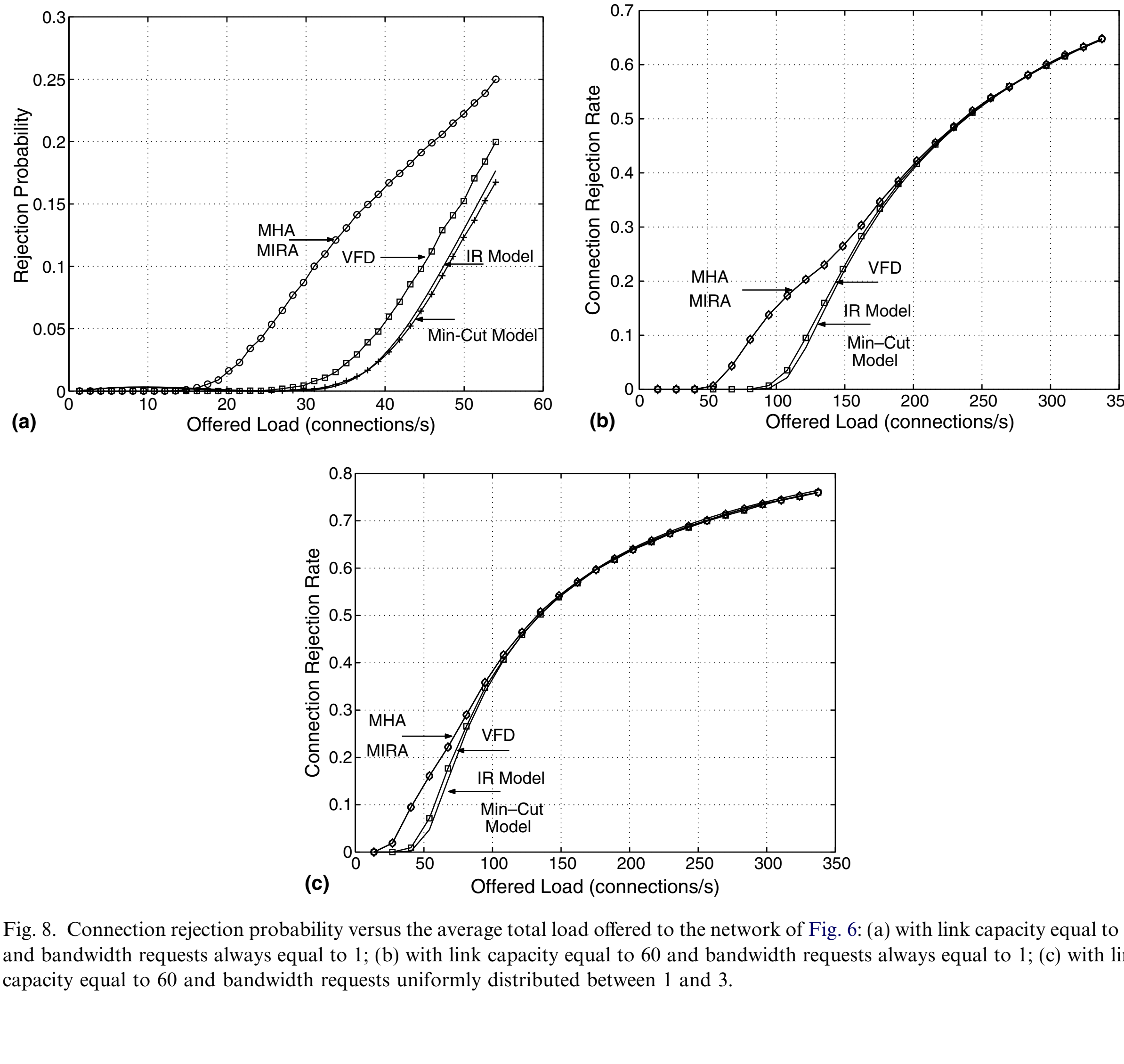

Fig. 8. Connection rejection probability versus the average total load offered to the network of Fig. 6: (a) with link capacity equal to 24 and bandwidth requests always equal to 1; (b) with link capacity equal to 60 and bandwidth requests always equal to 1; (c) with link capacity equal to 60 and bandwidth requests uniformly distributed between 1 and 3.

bound provided by the IR model when exact future connections requests are known.

To investigate the impact of connection lifetime distribution, we have considered a Pareto distribution with the same average as the previous exponential distribution and several shape parameters ($\alpha = 1.9, 1.95, 2.1, 3$). The performance observed in all cases are within 1% of those shown in Fig. 7.

To test the sensitivity of the performance to the network capacity, we have considered, for the network in Fig. 6, different parameters. The results, shown in Fig. 8, are very similar to those of Fig. 7. It is worthwhile to observe that in all the different scenarios considered the approximate model provides results very close to IR. This validates the approximate model that can be easily evaluated even in more complex networks.

The second network considered is shown in Fig. 9 where an equal traffic is offered at S_1 and S_2. All links have the same capacity equal to 120 and are bidirectional.

Also in this scenario VFD outperforms MIRA and MHA, as shown in Fig. 10. MHA is the worst due to its poor choice of the paths that are used until saturation before switching to other paths with less utilized links.

MIRA shows a performance that worsens as the offered load increases. In fact due to the critical links identified, $(0,1)$, $(0,2)$, $(0,3)$, $(1,4)$, $(2,4)$, $(3,4)$ for connections S_2–T_2 and $(1,0)$, $(1,2)$, $(1,4)$, $(0,3)$, $(2,3)$, $(4,3)$ for connections S_1–T_1, the path (1–2–3) is the only one available for connections S_1–T_1 and the path $(0,2,4)$ the only one for connections S_2–T_2.

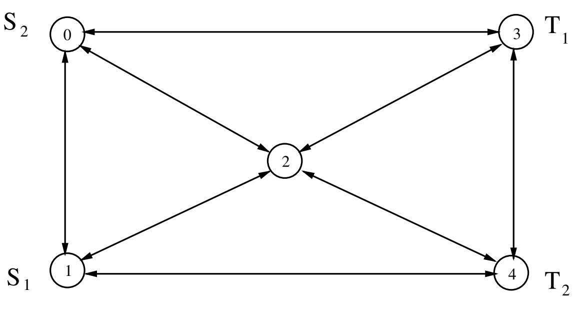

Fig. 9. Network topology with a large number of critical links.

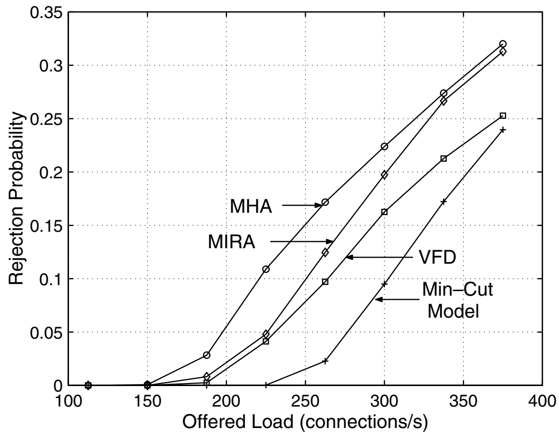

Fig. 10. Connection rejection probability versus the average total load offered to the network of Fig. 9.

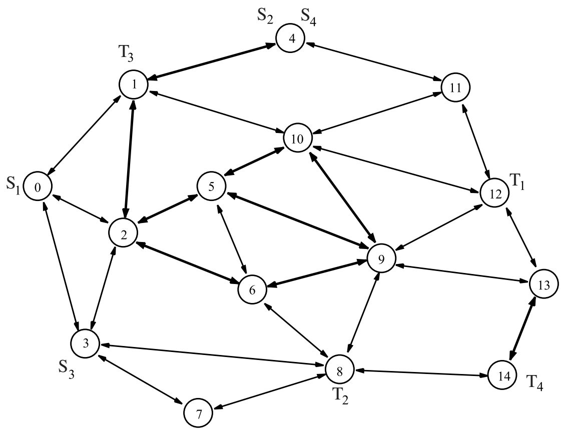

Fig. 11. Network topology with a large number of nodes, links, and source/destination pairs.

Due to computational complexity only the performance of the Min-Cut model is shown in Fig. 10.

A more realistic scenario that was first proposed in [14,15] is shown in Fig. 11. The links marked by heavy solid lines have a capacity of 480 while the others have a capacity equal to 120, in order to replicate the ratio between OC-48 and OC-12 links. The performance for the case of balanced offered traffic, considered in [14], are shown in Fig. 12.

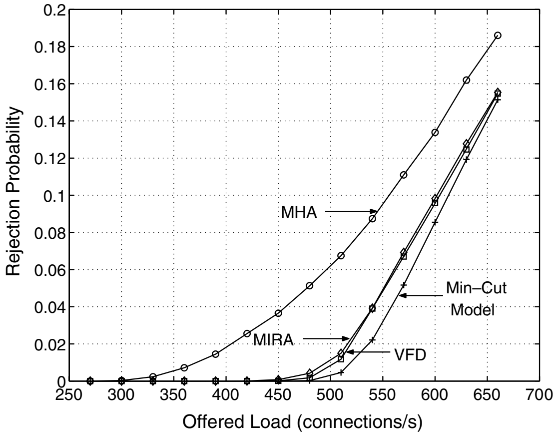

Fig. 12. Connection rejection probability versus the average total load offered to the network of Fig. 11.

VFD and MIRA achieve almost the same performance and are much better than MHA. VFD presents a slight advantage at low load since it starts rejecting connections at an offered load 10% higher than MIRA. We have measured that a rejection probability of 10^{-4} is reached at an offered load of 420 connections/s by MIRA as opposed to 450 connections/s for VFD. Also in this case the IR model is computationally too demanding. Therefore, we applied the Min-Cut model with the inversion algorithm proposed in [29], as the maximum multi-commodity flow is equal to 1200 bandwidth units.

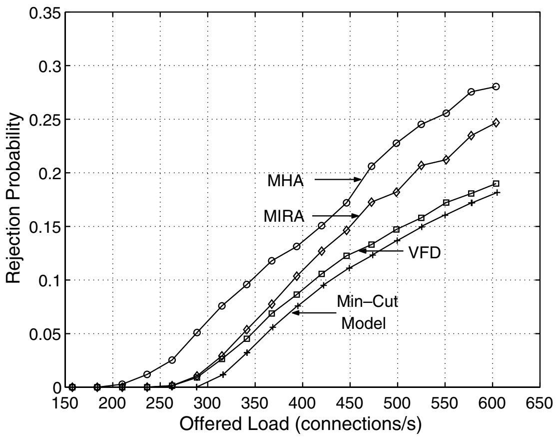

Fig. 13. Connection rejection probability versus the average total load offered to the network of Fig. 11, where the traffic between S_1–T_1 is four times higher than the traffic produced by the other pairs.

If we consider on the same topology an unbalanced load where for instance traffic S_1–T_1 is four times the traffic of the other sources, the improvement in the performance obtained by VFD is remarkable. The results shown in Fig. 13 confirm that unbalanced situations are more demanding on network resources and the rejection probability for the same given offered load is much higher. In these more critical network operating conditions VFD still well approaches the lower bound provided by the Min-Cut model.

6. Conclusions

We have discussed and analyzed the performance of online QoS routing algorithms for bandwidth guaranteed connections in MPLS and label switched networks.

To provide a theoretical bound on the performance achievable by dynamic online QoS routing algorithms we have proposed two novel mathematical models. The first is an Integer Linear Programming model that extends the well known maximum multi-commodity flow problem to include connections arrival-times and durations, while the second, which has a much lower complexity, is based on the application of the multi-class Erlang formula to a link with capacity equal to the residual capacity of the minimum network multi-commodity cut.

We have shown that the Virtual Flow Deviation scheme reduces the blocking probability with respect to previously proposed routing schemes and approaches the lower bounds provided by the mathematical models in the considered network scenarios.

References

[1] Hui-Lan Lu, I. Faynberg, An architectural framework for support of quality of service in packet networks, IEEE Communications Magazine 41 (6) (2003) 98–105.

[2] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol Label Switching Architecture, in: IETF RFC 3031, January 2001.

[3] L. Berger (Ed.), Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description, in: IETF RFC 3471, January 2003.

[4] A. Banerjee, L. Drake, L. Lang, B. Turner, D. Awduche, L. Berger, K. Kompella, Y. Rekhter, Generalized multiprotocol label switching: an overview of signaling enhancements and recovery techniques, IEEE Communications Magazine 39 (7) (2001) 144–151.
[5] S. Chen, K. Nahrstedt, An overview of quality-of-service routing for the next generation high-speed networks: problems and solutions, IEEE Network 12 (6) (1998) 64–79.
[6] J.L. Marzo, E. Calle, C. Scoglio, T. Anjali, QoS online routing and MPLS multi-level protection: a survey, IEEE Communications Magazine 41 (10) (2003) 126–132.
[7] Bin Wang, Xu Su, C.L.P. Chen, A new bandwidth guaranteed routing algorithm for MPLS traffic engineering, in: IEEE International Conference on Communications, ICC 2002, vol. 2, 2002, pp. 1001–1005.
[8] Z. Wang, J. Crowcroft, QoS routing for supporting resource reservation, IEEE Journal on Selected Areas in Communications (September) (1996).
[9] A. Orda, Routing with end to end QoS guarantees in broadband networks, in: IEEE–INFOCOM'98, March 1998.
[10] B. Awerbuch et al., Throughput competitive on line routing, in: 34th Annual Symposium on Foundations of Computer Science, Palo Alto, CA, November 1993.
[11] S. Suri, M. Waldvogel, D. Bauer, P.R. Warkhede, Profile-based routing and traffing engineering, Computer Communications 26 (4) (2003) 351–365.
[12] D.O. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: Extensions to RSVP for LSP tunnels, in: IETF RFC 3209, December 2001.
[13] R. Guerin, D. Williams, A. Orda, QoS routing mechanisms and OSPF extensions, in: Proceedings of Globecom, 1997.
[14] Murali S. Kodialam, T.V. Lakshman, Minimum interference routing with applications to MPLS traffic engineering, in: Proceedings of INFOCOM, pp. 884–893, 2000.
[15] Koushik Kar, Murali Kodialam, T.V. Lakshman, Minimum interference routing of bandwidth guaranteed tunnels with MPLS traffic engineering applications, IEEE Journal on Selected Areas in Communications 18 (12) (2000).
[16] A. Capone, L. Fratta, F. Martignon, Dynamic routing of bandwidth guaranteed connections in MPLS networks, International Journal on Wireless and Optical Communications 1 (1) (2003) 75–86.
[17] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows, Prentice-Hall, 1993.
[18] ILOG CPLEX. Available at: <http://www.ilog.com/products/cplex/>.
[19] AMPL: A Modeling Language for Mathematical Programming. Available at: <http://www.ampl.com>.
[20] J.F.P. Labourdette, G.W. Hart, Blocking probabilities in multitraffic loss systems: insensitivity, asymptotic behavior, and approximations, IEEE Transactions on Communications 40 (1992) 1355–1366.
[21] Tom Leighton, Satish Rao, Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms, Journal of the ACM 46 (6) (1999).
[22] Y. Aumann, Y. Rabani, An O($\log k$) approximate min-cut max-flow theorem and approximation algorithm, SIAM Journal on Computing 27 (1) (1998) 291–301.
[23] R. Guerin, H. Ahmadi, M. Naghshineh, Equivalent capacity and its application to bandwidth allocation in high speed networks, IEEE Journal on Selected Areas in Communications (September) (1991) 968–981.
[24] J.A. Schormans, J. Pitts, K. Williams, L. Cuthbert, Equivalent capacity for on/off sources in ATM, Electronic Letters 30 (21) (1994) 1740–1741.
[25] L. Fratta, M. Gerla, L. Kleinrock, The flow deviation method: An approach to store-and-forward network design, Networks 3 (1973) 97–133.
[26] D. Bertsekas, R. Gallager, Data Networks, Prentice-Hall, 1987.
[27] J.S. Kaufman, Blocking in a shared resource environment, IEEE Transactions on Communications 29 (10) (1981) 1474–1481.
[28] A.A. Nilsson, M. Perry, A. Gersht, V.B. Iversen, On multirate Erlang-B computations 16th International Teletraffic Congress; ITC'16, Elsevier Science, 1999, pp. 1051–1060.
[29] G.L. Choudhury, K.K. Leung, W. Whitt, An inversion algorithm to compute blocking probabilities in loss networks with state-dependent rates, IEEE/ACM Transactions on Networking 3 (5) (1995) 585–601.

Antonio Capone received the Laurea degree (MS degree equivalent) and the PhD degree in telecommunication engineering from the Politecnico di Milano in July 1994 and June 1998, respectively. In 2000, he was a visiting scientist at the University of California, Los Angeles. He is now an associate professor in the Department of Electronics and Information at the Politecnico di Milano. His current research activities include packet access in wireless cellular network, routing and MAC for multihop wireless networks, congestion control and QoS issues of IP networks, network planning and optimization. He is a member of the IEEE and the IEEE Communications and Vehicular Technology Societies.

Luigi Fratta received the doctorate degree in electronics engineering from the Politecnico di Milano, Italy, in 1966. From 1967 to 1970, he worked at the Laboratory of Electrical Communications, Politecnico di Milano. As a research assistant in the Computer Science Department, University of California, Los Angeles (UCLA), he participated in data network design under the ARPA project from 1970 to 1971. From November 1975 to September 1976, he was at the

Computer Science Department of IBM Thomas J. Watson Research Center, Yorktown Heights, New York, working on modeling analysis and optimization techniques for teleprocessing systems. In 1979, he was a visiting associate professor in the Department of Computer Science at the University of Hawaii. In the summer of 1981, he was at the Computer Science Department, IBM Research Center, San Jose, California, working on local area networks. During the summers of 1983, 1989, and 1992, he was with the Research in Distributed Processing Group, Department of Computer Science, UCLA, working on fiber optic local area networks. During the summer of 1986, he was with Bell Communication Research working on metropolitan area networks. In 1994, he was a visiting scientist at NEC Network Research Laboratory, Japan. Since 1980, he has been a full professor in the Dipartimento di Elettronica e Informazione at the Politecnico di Milano. His current research interests include computer communication networks, packet switching networks, multiple access systems, modeling and performance evaluation of communication systems, local area networks, wireless cellular systems, and integrated services over IP networks. Dr. Fratta is a fellow of IEEE.

Fabio Martignon received the Laurea and the PhD degree in telecommunication engineering from the Politecnico di Milano in October 2001 and May 2005, respectively. He is now an assistant professor in the Department of Management and Information Technology at the University of Bergamo. His current research activities include routing for multihop wireless networks, congestion control and QoS routing over IP networks.