International Journal on Wireless & Optical Communications
Vol. 1, No. 1 (2003) 75–86

World Scientific
www.worldscientific.com

DYNAMIC ROUTING OF BANDWIDTH GUARANTEED CONNECTIONS IN MPLS NETWORKS

ANTONIO CAPONE*, LUIGI FRATTA†, and FABIO MARTIGNON‡
DEI, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milan, Italy
**capone@elet.polimi.it*
†fratta@elet.polimi.it
‡martignon@elet.polimi.it

Received May 2003

Finding a path in the network for a new incoming connection able to guarantee some quality parameters such as bandwidth and delay is the task of QoS routing techniques developed for new IP optical networks based on label forwarding techniques such as Generalized Multiprotocol Label Switching (GMPLS). In this paper we focus on the routing of bandwidth guaranteed flows in a dynamic scenario where new connection requests arrive at the network edge nodes. When more than one path satisfying the bandwidth demand exists, the selection of the path aims at minimizing the blocking probability of future requests. We propose a new routing algorithm named Virtual Flow Deviation (VFD) which exploits the information of the ingress and egress nodes of the network and the traffic statistics. This new algorithm, in most scenarios, achieves remarkable reduction in the blocking probability with respect to previously proposed schemes.

Keywords: QoS routing; MPLS; flow deviation.

1. Introduction

The development of the Internet has been impressive in the recent years. The upcoming high-speed optical networks are expected to support a wide variety of communication-intensive real-time multimedia applications. However, the current Internet architecture offers mainly a best-effort service and does not meet the requirements of future integrated services networks that will be designed to carry heterogeneous data traffic [Chen & Nahrstedt, 1998].

In order to offer guaranteed end-to-end performance (in terms of bounded delay, jitter or loss rate), it is necessary to introduce some sort of resource reservation mechanism in the Internet. With classical IP routing, packet forwarding is performed independently at each router in the network and is based only on the destination address carried in the packet. Classical IP routing policy selects the shortest path to the destination by exploiting distributed routing protocols. Hence, when insufficient resources are available on the shortest path, the quality degrades.

Recently, substantial effort has been spent to evolve conventional IP routing architecture and protocols by providing them with additional functionalities using the Multi Protocol Label Switching method (MPLS) [Rosen *et al.*, 2001]. One of the key aspects of MPLS is the addition of a new connectivity abstraction: explicitly routed point-to-point paths can be established using label-based forwarding mechanisms.

Recent work has extended and adapted the MPLS control plane, and in particular the MPLS constraint-based routing, so that it can be used not just with electronic MPLS switches, but also with optical crossconnects (OXCs) [Awduche *et al.*, 2002]. This is a fundamental step in the evolution and integration of data and optical network architectures. Some enhancements are clearly required for the existing MPLS routing and signaling protocols to address the peculiar characteristics of optical transport networks. These protocol extensions are being standardized by the Internet Engineering Task Force (IETF) under the framework of Generalized Multiprotocol Label Switching (GMPLS) [Baerjee *et al.*, 2001].

GMPLS is a multipurpose control plane paradigm that provides the necessary bridges between the IP and photonic layers as it supports either devices that perform only packet switching, as well as devices that perform switching in the time, wavelength and space domains.

GMPLS extensions to the original MPLS framework can be summarized as follows:

- Enhancements to the RSVP-TE and CR-LDP signaling protocols to allow the signaling and instantiation of optical channel trails in optical transport networks and other connection-oriented networking environments [Ashwood-Smith & Berger, 2002];
- A new link management protocol, LMP, designed to address the issues related to link management in optical networks [Ashwood-Smith & Berger, 2002];
- Enhancements to OSPF and IS-IS interior gateway protocols (IGPs) to advertise the availability of optical resources in the network (i.e. residual bandwidth on wavelengths, interface type) and other network attributes and constraints [Kompella & Rekhter, 2001].

These enhancements proposed for GMPLS allow per flow path selection and Quality of Service parameters to be taken into account by the routing algorithm. The notion of Quality of Service (QoS) has been introduced to capture the qualitatively and quantitatively defined performance contract between the service provider and the user applications. The QoS requirement of a connection can be given as a set of link constraints. Such contraints can be expressed, for instance, as bandwidth constraints specifying that the path selected for the connection of the requesting user has sufficient bandwidth to meet the connection requirement.

The goal of QoS routing algorithms is twofold:

- satisfying the QoS requirements for every admitted connection;
- achieving global efficiency in resource utilization.

In this paper, we focus on the problem of QoS routing. First of all, we review some of the proposed QoS routing algorithms, such as the Min-Hop Algorithm (MH) [Awduche *et al.*, 1999], the Widest Shortest Path Algorithm (WSP) [Guerin *et al.*, 1997] and the Minimum Interference Routing Algorithm (MIRA) [Kodialam & Lakshman, 2000]. We will describe in some detail MIRA, which has more features compared to other algorithms, as it takes explicitly into account the topology position of the ingress and egress points of the network, i.e. the routers through which the traffic enters and exits the network.

We analyze and compare their performance based on results obtained under a variety of simulated scenarios. We observe that these algorithms are unable to achieve good performance when the traffic statistics at each ingress point are different (for instance, when an ingress node offers to the network a traffic significantly higher than other nodes). Furthermore, all the proposed algorithms fail to consider traffic statistics which can be easily measured at each ingress node.

To overcome these limitations, we propose a new QoS routing algorithm, called the Virtual Flow Deviation (VFD), which takes into account the information of the ingress and egress nodes of the network and the traffic statistics. More precisely, VFD exploits the knowledge of the position of the ingress/egress nodes of the network, and uses the statistics information about the traffic offered to the network through each ingress point in order to forecast future connection arrivals. For every connection request, VFD creates a set of *virtual calls* based on the observed traffic statistics. These virtual calls represent the calls which are likely to request resources to the network in the immediate future, and will thus interfere with the current one. In order to improve the global resource utilization, VFD routes the current call together with the virtual calls using the Flow Deviation method [Fratta *et al.*, 1973].

In order to assess the effectiveness of the proposed scheme, we analyze the performance of VFD under a variety of scenarios, and we compare it with that achieved by existing routing algorithms.

The paper is structured as follows: in Sec. 2 we address the QoS routing problem and some existing routing algorithms. In Sec. 3 we introduce the Virtual Flow Deviation algorithm. In Sec. 4 we discuss and compare the performance of these algorithms under a variety of simulated scenarios. Finally, Sec. 5 concludes the paper.

2. QoS Routing

The concept of Quality of Service (QoS) involves a performance trade-off between the Internet Service Provider (achieving high utilization) and the hosts' applications (perceiving high throughput and low delay). The QoS constraints parameters of a single connection are usually specified in terms of minimum guaranteed bandwidth, maximum tolerable delay and/or jitter and maximum tolerable loss rate.

The main goal of a QoS routing technique is to determine a path that can guarantee the constraints requested by the incoming connection and reject as few connections as possible.

In this paper we focus only on bandwidth guaranteed paths, and we assume that all the other quality parameters can be controlled defining an equivalent flow bandwidth in a proper way [Guerin *et al.*, 1991; Schormans *et al.*, 1994].

Let's model a network as a graph (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 single connection requires a path from S_i to T_i. Each link (i, j) has associated the capacity C_{ij} and the actual flow F_{ij}. The residual bandwidth 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.

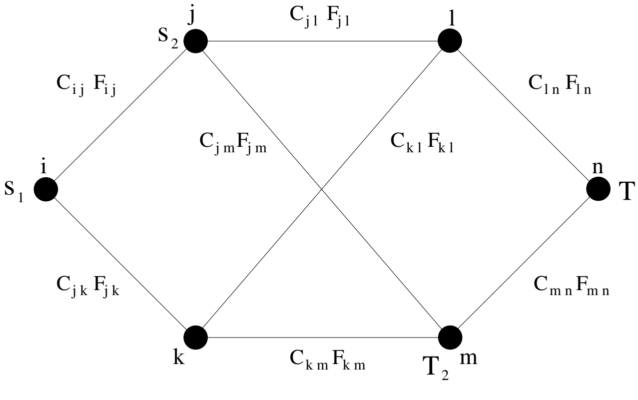

Fig. 1. QoS network state.

Referring to a new connection with requested bandwidth d_k, a link is defined as *feasible* if $R_{ij} \geq d_k$. 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 maximum residual bandwidth of that path.

In the following we review some of the algorithms available in the literature.

2.1. *Min-Hop Algorithm*

The Min-Hop Algorithm (MHA) [Awduche *et al.*, 1999] routes an incoming connection along the path which reaches the destination node using the minimum number of feasible links.

This scheme, based on the Dijkstra algorithm, is simple and computationally efficient. However, using MHA can result in heavily loaded bottleneck links in the network, as it tends to overload some links leaving others underutilized. The cost given to each link, in fact, remains unvaried and independent of the current link load and therefore MHA tends to use the same paths until saturation is reached before switching to other paths with underutilized links.

2.2. *Widest Shortest Path Algorithm*

The Widest Shortest Path Algorithm (WSP), proposed in [Guerin *et al.*, 1997], is an improvement of the Min-Hop algorithm, as it attempts to load-balance the network traffic. In fact, WSP chooses a feasible path with minimum hop count and, if there are a multiple of 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 which are used until saturation before switching to other feasible paths.

2.3. *Minimum Interference Routing Algorithm*

The Minimum Interference Routing Algorithm (MIRA), proposed in [Kodialam & Lakshman, 2000], explicitly takes into account the location of the ingress and egress routers. The key idea of MIRA is to route an incoming connection over a path which least interferes with possible future requests.

Specifically, an incoming connection request between (S_i, T_i) is routed with the goal of maximizing an objective function which is either the minimum maximum-flow (maxflow) of all *other* ingress-egress pairs or a weighted sum of maxflows, where weights α_{ST} assigned to each ST pair reflect the "importance" of the flow.

In order to achieve an on-line routing algorithm, MIRA keeps an updated list of the *critical* links, i.e. the links whose use by the incoming call diminishes the maxflow 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) *other* than (S_i, T_i). The weight w of each link l is then set according to the equation $w(l) = \sum_{(S,T):l \in L_{ST}} \alpha_{ST}$, and the route which causes the minimum interference to other source/destination pairs is selected.

In spite of its more sophisticated functions, MIRA still has the following limitations whose effects will be shown in the discussion of numerical results:

- MIRA discourages the use of critical links based only on the number of other S-T pairs which could use them, without verifying if these S-T pairs actually use these links. Evidently, if one of these other S-T pairs introduces a low traffic in the network, the *criticality* of the links which diminish its maxflow is far less important than that of S-T pairs which produce a large amount of traffic. As a consequence, MIRA preserves the use of certain links which remain underutilized, 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 maxflows. However, in [Kodialam & Lakshman, 2000] the weights are chosen offline and they do not adapt to the changes in network traffic. Hence this solution does not provide the flexibility necessary to an on-line routing scheme.
- In its on-line implementation, MIRA sets the link weights almost in a static way according only to their level of criticality. In fact, the only event which can cause the redistribution of new weights is the saturation of some links, similarly to the Min-Hop algorithm.
- While chosing a path for an incoming request, MIRA does not take into account how the new call will affect the future requests of the *same* ingress/egress pair (auto-interference).

3. Virtual Flow Deviation

In the previous sections we have summarized the features and the limitations of the existing QoS routing algorithms. In this section we propose a new technique, called the Virtual Flow Deviation (VFD), which aims to overcome these limitations by exploiting all the information available which has not been utilized, or even considered, by the other routing algorithms.

To better describe the current state of the network and to forecast its future state one can add to the topological information on the location of ingress/egress pairs, used by MIRA, the traffic statistics obtained by measuring the load offered to the network at each source node. This information plays a key role in deciding how to route incoming requests to prevent network congestion.

By exploiting the knowledge of the offered traffic, we can forecast how many new connections will probably be generated at each S-T pair in the immediate future. These new calls, which are likely to be offered to the network, will interfere with the current call to be routed, and they should thus be considered in the routing process.

In order to take into account the future traffic offered to the network, VFD routes not only the real call, but also a certain number of *virtual* calls which represent an estimate (based on measured traffic statistics) of the connection requests that will probably interfere with the current, real call. The number of these virtual calls, as well as the origin and the bandwidth requested should reflect as closely as possible the real future network conditions. Hence, we determine these parameters based on the past traffic statistics of the various ingress/egress pairs, which we will explain in detail in the next subsection. With this mechanism, we can take into account both the interference and the auto-interference produced by the current and future calls between every S-T pair.

The precision of the measured traffic statistics is an important factor for the performance of the Virtual Flow Deviation algorithm. However, even if the traffic statistics are not measured with high accuracy, they possess a richer information than the topological information about the position of source and destination nodes exploited by MIRA.

All the information concerning network topology and estimated offered load must be used to produce a path selection which uses at the best the network resources and minimizes the number of rejected calls. Such a path selection is performed in VFD by the Flow Deviation method, which allows us to determine the optimal routing of all the flows entering the network through all the different source/destination pairs.

Before describing in detail the VFD algorithm, we give a high-level scheme of its functionalities.

3.1. *The virtual calls*

Each call offered to the network, either real or virtual, is represented by the notation (S, T, d), where S and T are the source and destination nodes of the call, respectively, and d is its bandwidth requirement. The determination of these three parameters for each virtual call is quite critical, as they must reflect as close as possible the dynamic traffic demand placed on the network. More precisely, we have to determine how many virtual calls should be generated, their source/destination pairs, and their bandwidth request. In this process, we can easily measure and distribute to each S-T pair the two following parameters:

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

Note that, for sake of simplicity, we have considered multiple bandwidth requests of a given bandwidth unit. However, the algorithm works also with bandwidth requirements which can assume any real value.

If we define the total average load offered to the network, Λ, as:

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

we can evaluate the probability P_{S_i,T_i} to receive a call between the node pair (S_i, T_i) as $P_{S_i,T_i} = \frac{\lambda_{S_i,T_i}}{\Lambda}$, while the probability of having a request of d bandwidth units at the ith source node, P_{d_i}, is estimated by $P_{d_i} = \frac{n_d}{N}$.

The parameters (S_i, T_i, d_i), which completely determine the virtual calls, are generated by extracting random values according to the probability density functions P_{S_i,T_i} and P_{d_i}, derived as described above. The virtual calls are routed together with the real call, represented by (S_R, T_R, d_R), using the Flow Deviation method to ensure an optimal flow assignment.

To determine the number N_v of virtual calls which must be generated we have considered two different approaches as listed below.

The first one takes into account the variations in the total load offered to the network, and estimates the average number $\bar{N}$ of calls routed (active calls) in the network over the past T seconds. When the new call request arrives, $N_v = \lfloor(\bar{N} - N_A)\rfloor$ virtual calls are generated, where N_A the current number of active calls. Note that if $N_A > \bar{N}$, no virtual call is generated.

The second approach is based on the maximum number of active calls routed in the network, $N_{\max}$, and the number of virtual calls is given by $N_v = \lfloor(N_{\max} - N_A)\rfloor$. The underlying assumption in this approach is to consider the network operating close to its saturation. This condition, which stresses the effectiveness of the routing algorithm, is useful when comparing performance. All the numerical results presented in Sec. 4 have been derived by implementing this second approach.

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 the process for generating N_v associated virtual calls, as described in the previous section, is activated. 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. This step is repeated until all calls have been considered. The procedure stops if the real call cannot be routed.

In Step 2 the routing of all ACTIVE calls is optimized using the Flow Deviation Method [Fratta *et al.*, 1973; Bertsekas & Gallager, 1987] which will be described in the next section.

Thereafter step one is repeated for the NON ACTIVE calls. If at least one NON ACTIVE call is declared ACTIVE Step 2 is repeated and the procedure is iterated until either all calls are ACTIVE or Step 1 does not define any new call as ACTIVE.

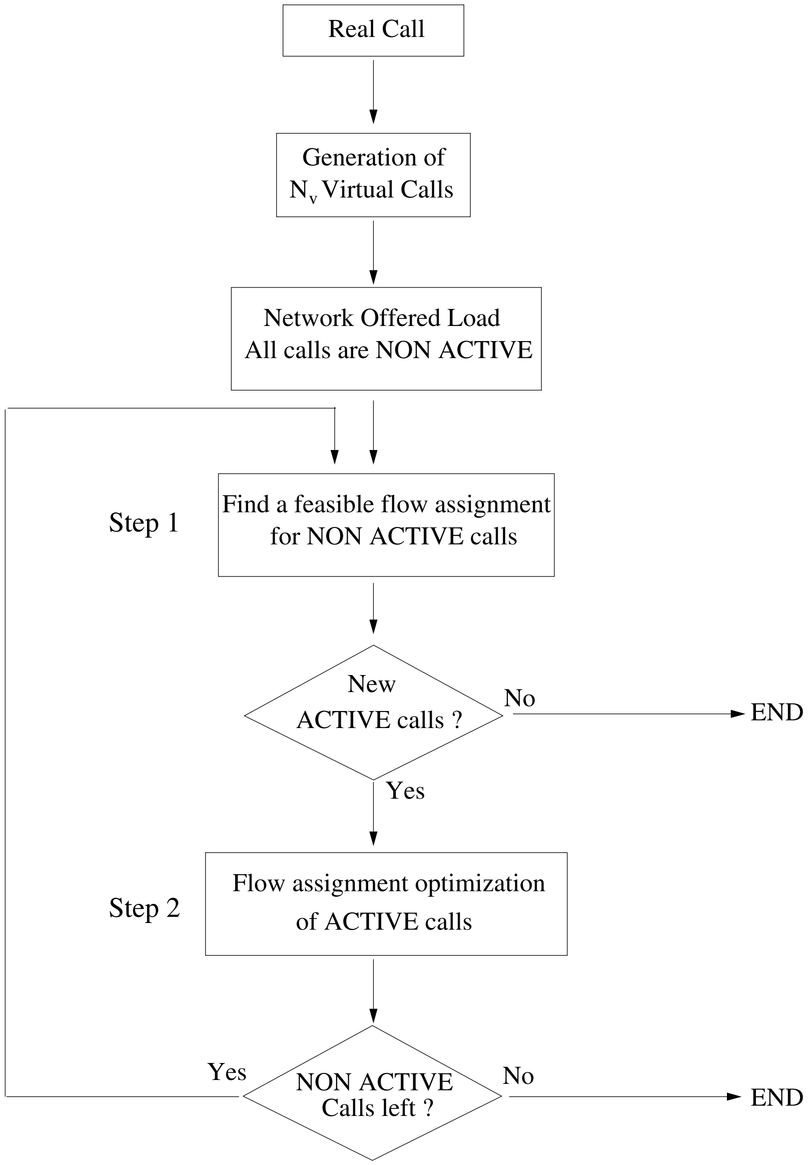

Fig. 2. The Virtual Flow Deviation algorithm.

At the end of the procedure the real call has been routed on an optimal path considering an expected future evolution of the network traffic load.

The feasible flow assignment is obtained in Step 1 by using the Shortest Path Algorithm (Dijkstra) applied to the network whose link 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. A flag to identify ACTIVE and NON ACTIVE connections has been added in the connection description.

3.3. *The flow deviation method*

In this section we briefly review the Flow Deviation Method [Fratta *et al.*, 1973], which allows us to determine the optimal routing of all the flows entering the network at different source/destination pairs.

It has been shown that optimal routing directs traffic exclusively along paths which are shortest with respect to some link lengths that depend on the flows carried by the links [Bersekas & Gallager, 1987]. Consequently, the optimal routing results only if flows travel along the minimum first derivative length (MFDL) paths for each source/destination pair. Equivalently, a routing is

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:
            $w_{ij} = \frac{1}{C_{ij}-F_{ij}}$ if $F_{ij} < C_{ij}$
            $w_{ij} = \infty$ 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)
```

strictly suboptimal only if there is a positive amount of flow that travels on a non-MFDL path. This suggests that suboptimal routing can be improved by shifting the flow, for each source/destination pair, from any path to a MFDL path.

The Flow Deviation method (FD) implements this idea. For a more formal description of the method, we introduce the following notation:

W : the set of all the source/destination pairs;
d_w : the traffic input of the source/destination pair $w \in W$;
P_w : the set of all the directed paths which connect the source/destination pair $w \in W$;
x_p : the flow assigned to the path p.

Given:

- The network topology with the link capacity matrix $C = \{C_{ij}\}$
- The traffic matrix $\Gamma = \{\gamma_{ij}\}$

the Flow Deviation method minimizes the average total network delay D (based on M/M/1 approximations) by routing every flow over a MFDL path. The objective function minimized by the Flow Deviation Method is hence:

$$D = \frac{1}{\gamma} \sum_{(i,j)} \frac{F_{ij}}{C_{ij} - F_{ij}} , \qquad (1)$$

where F_{ij} represents the flow routed over the link ij and γ is the total input flow into the network.

The Flow Deviation method starts from a given feasible path flow vector $x = \{x_p\}$ and finds a minimum first derivative length (MFDL) path for each source/destination pair. The flow assignment is incrementally changed along the descent direction of the objective function.

To achieve this goal, the Flow Deviation method sets the weight W_{ij} of the link ij as the partial derivative of D with respect to the flow rate which is traversing the link ij, F_{ij}, evaluated at the current flow assignment (x):

$$W_{ij} = D'(F_{ij}) = \frac{\delta D}{\delta F_{ij}} = \frac{C_{ij}}{\gamma (C_{ij} - F_{ij})^2} . \qquad (2)$$

Thereafter, the new flow assignment is determined by using the shortest path algorithm in terms of the W_{ij}.

Let $\bar{x} = \{\overline{x_p}\}$ be the vector of path flows that would result if all input d_w for each source/destination pair $w \in W$ is routed along the corresponding MFDL path. Also, we let α^* be the stepsize that minimizes $D[x+\alpha(\bar{x}-x)]$ among every possible choice of $\alpha \in [0, 1]$, i.e.

$$D[x + \alpha^*(\bar{x} - x)] = \min_{\alpha \in [0,1]} D[x + \alpha(\bar{x} - x)] .$$

The new set of path flows is obtained by:

$$x_p := x_p + \alpha^*(\overline{x_p} - x_p) , \quad \forall\, p \in P_w,\ w \in W$$

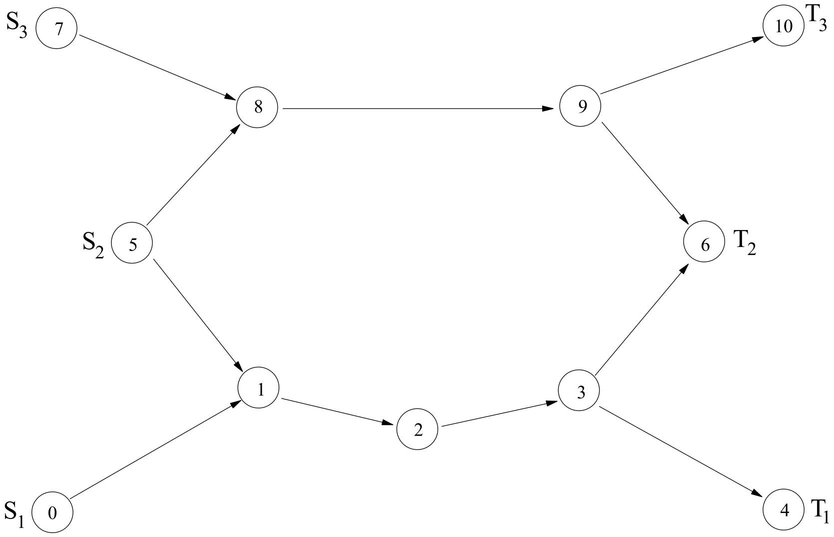

Fig. 3. Network topology with unbalanced offered load: the source/destination pairs S_2-T_2 and S_3-T_3 offer to the network a traffic load which is four times higher than that offered by the pair S_1-T_1.

and the process is then iterated. By incrementally changing the previous flow assignment into the new one, the optimal flow assignment is determined. This procedure converges to the optimum as the objective function (1) and its first derivative are monotonical non-decreasing functions.

The Virtual Flow Deviation algorithm uses a slightly modified version of the Flow Deviation method, in which the flows cannot be split across multiple routes. Therefore, the variable α in each step can take only discrete values in $[0, 1]$, to guarantee that no flow is split over more than one path.

4. Numerical Results

In this section we compare the performance of the Virtual Flow Deviation algorithm with that of the Min-Hop Algorithm and the MIRA by referring to three different network scenarios in order to cover a wide range of possible environments. The performance function we consider is the percentage of rejected calls versus the average total load offered to the network.

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

In this simple topology only one path is available to route connections between S_1-T_1 and S_3-T_3, while connections S_2-T_2 can choose between two different paths.

This case shows the main limitation of MIRA which does not consider the information about the total load offered to the network. Since the links $(1, 2)$, $(2, 3)$ and $(8, 9)$ are critical for S_2-T_2, the route selected by MIRA follows the path with the minimum number of critical links (5-8-9-6 in the example). Unfortunately this interferes with the path (7-8-9-10) that carries the high load of S_3-T_3. This choice will penalize the performance as shown in Fig. 4.

VFD achieves the best performance since it exploits the information on the unbalanced load. The performance of MHA and MIRA are exactly the same. In fact MIRA operates for the connections between S_2-T_2 the same path selection of MHA, since the path (5-8-9-6) is shorter than (5-1-2-3-6).

In order to compare how the distribution of connection lifetimes impacts the performance of VFD, we have considered the same scenario with connections having a Pareto distribution of the lifetimes with the same average duration as before as well as various shape parameters ($\alpha = 1.9, 1.95, 2.1, 3$). We observed that the performance of all the routing algorithms almost overlaps (less than 1% of variation)

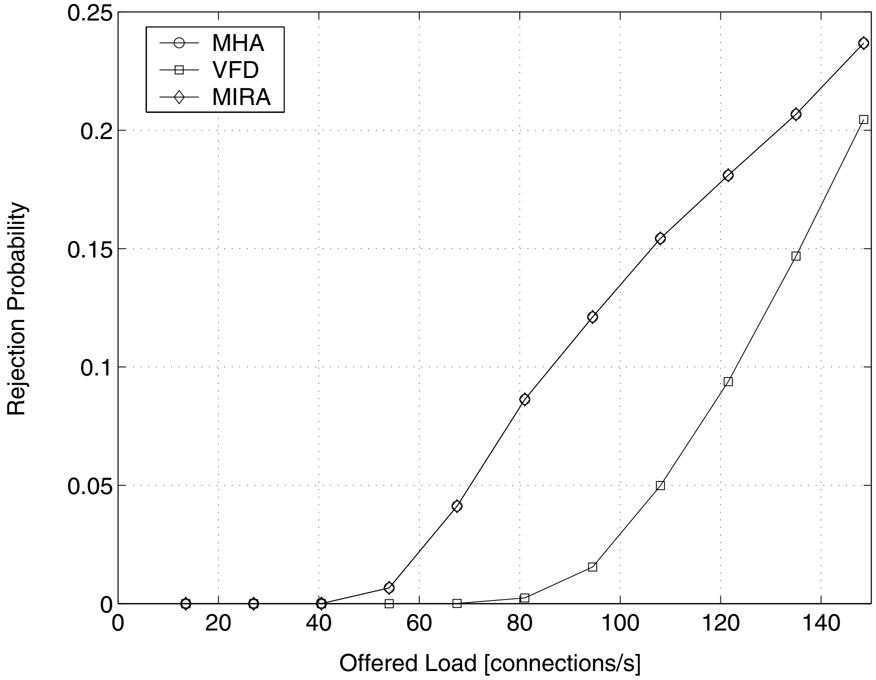

Fig. 4. Connection rejection probability versus the average total load offered to the network of Fig. 3.

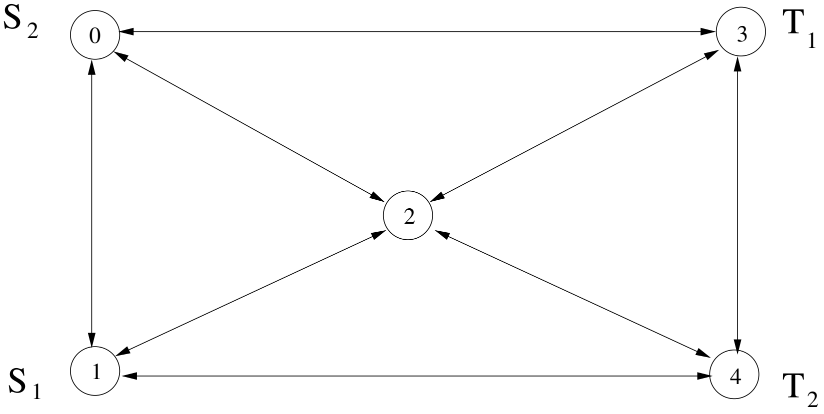

Fig. 5. Network topology with a large number of critical links.

with that obtained with exponentially distributed connections lifetimes as shown in Fig. 4.

The second network considered is shown in Fig. 5 where a balanced traffic is offered at S_1 and S_2. All links have the same capacity (120 bandwidth units) and are bidirectional.

The critical links identified by MIRA are $(0,1)$, $(0,2)$, $(0,3)$, $(1,4)$, $(2,4)$, $(3,4)$ 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. This results in only path (1-2-3) being available for connections S_1-T_1 and the path $(0,2,4)$ available for connections S_2-T_2.

This is a very limiting way of operation that penalizes MIRA. As shown in Fig. 6, VFD can reach a more balanced routing using all the available paths with no limitation.

In the third scenario we have considered the network shown in Fig. 7 that was proposed in [Kodialam & Lakshman, 2000] which represents a more realistic scenario. All links are bidirectional. Those marked by heavy solid lines have a capacity of 480 bandwidth units while the others have a capacity equal to 120 bandwidth units, in order to model the capacity ratio of OC-12 and OC-48 links. The

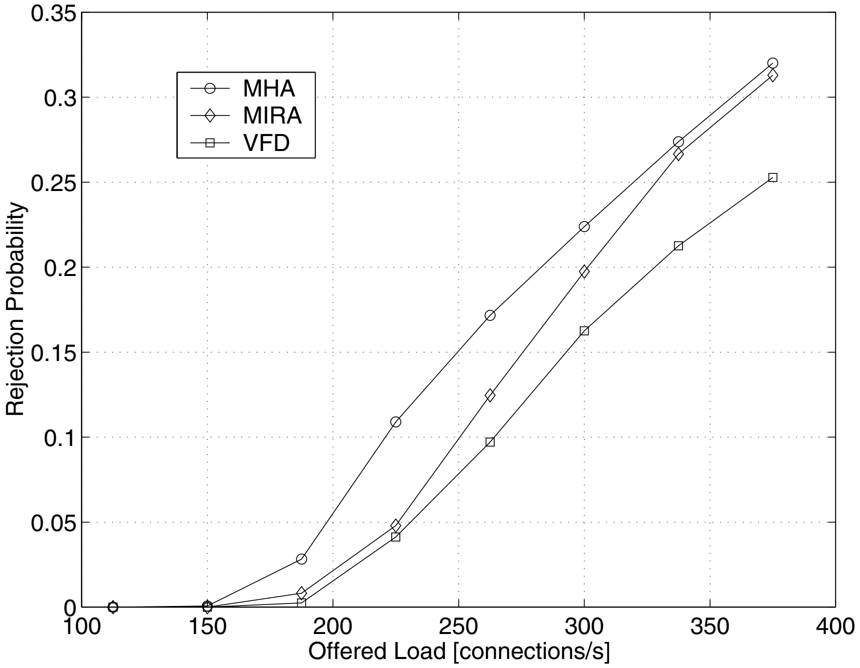

Fig. 6. Connection rejection probability versus the average total load offered to the network of Fig. 5.

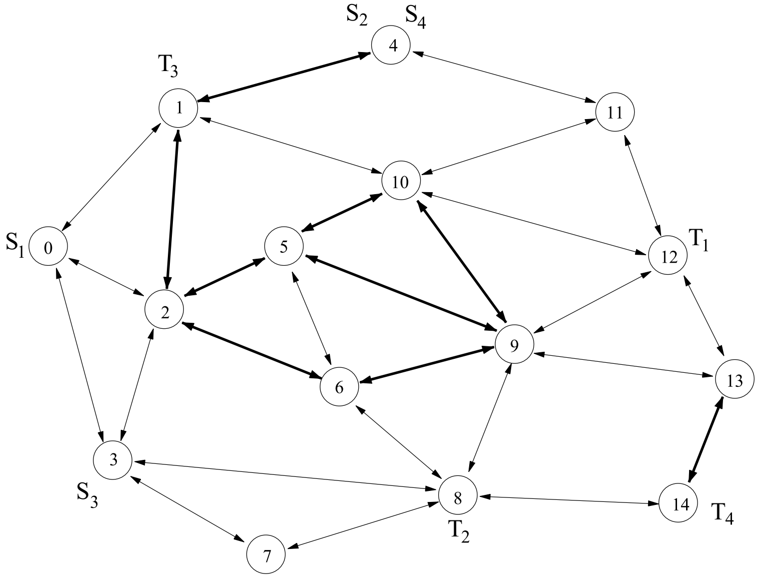

Fig. 7. Network topology with a large number of nodes, links, and source/destination pairs.

performance for the case of balanced offered traffic, considered in [Kodialam & Lakshman, 2000], are shown in Fig. 8.

VFD and MIRA achieve almost the same performance. However, VFD presents some advantages at low rejection probabilities 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.

If we consider on the same topology an unbalanced load where for instance the traffic S_1-T_1 is four times the traffic of the other sources, the improvement in the performance obtained by VFD is much more significant. The curves shown in Fig. 9

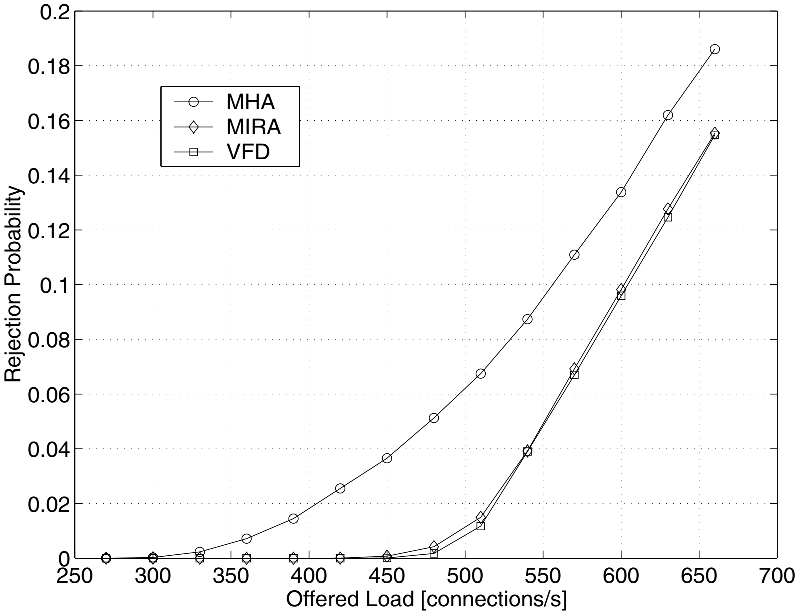

Fig. 8. Connection rejection probability versus the average total load offered to the network of Fig. 7.

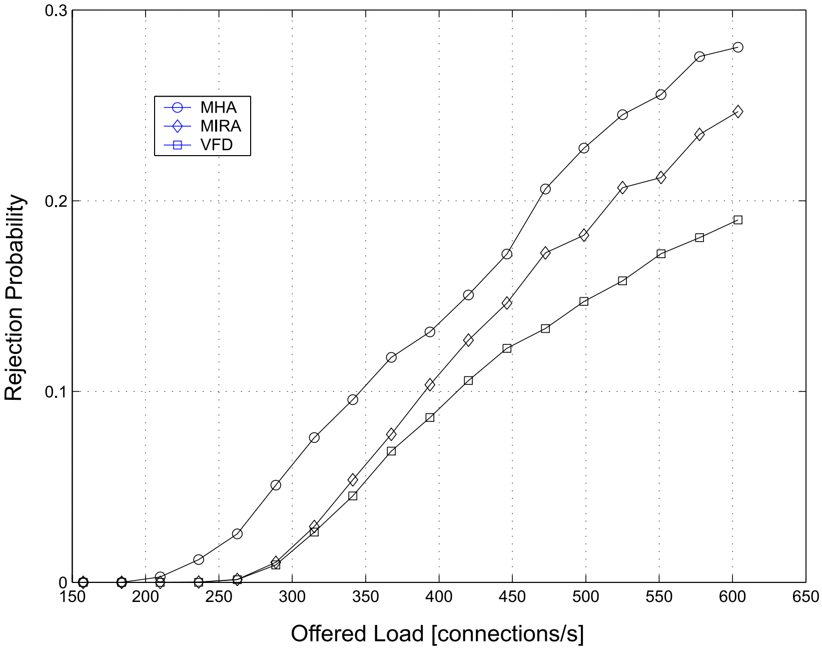

Fig. 9. Connection rejection probability versus the average total load offered to the network of Fig. 7, where the traffic between S_1-T_1 is four times higher than the traffic produced by the other pairs.

confirm that the unbalanced situations are more demanding on network resources with respect to the balanced case as the rejection probability for the same given offered load is much higher. In these more critical network operations VFD has proven to be more effective in providing improvements of the order of 20%.

5. Conclusions

We have proposed the Virtual Flow Deviation (VFD), a new algorithm for explicit-routing of bandwidth guaranteed connections in MPLS and GMPLS networks. VFD exploits the information about the ingress and egress nodes of the network and the traffic statistics.

As a key innovation with respect to existing QoS routing algorithms, VFD performs a path selection based not only on the current state of the network, but also on an estimate of its future evolution. This goal is achieved by routing a set of *virtual* calls together with the current call using the Flow Deviation algorithm to ensure an optimal disposition of all the flows. Virtual calls represent an estimate (based on traffic statistics measured at each ingress node) of the calls which are likely to be offered to the network during the current connection lifetime, thus interfering with it.

VFD allows us to achieve lower connection rejection rates as compared to the existing algorithms, especially in the more critical network operations with unbalanced traffic offered at the ingress nodes.

We have shown that this new algorithm allows us to reduce remarkably the blocking probability in most scenarios with respect to previously proposed schemes.

References

Ashwood-Smith, P. and Berger, L. [2002] "Generalized MPLS signaling — CR-LDP extensions," in *Internet Draft*, *draft-ietf-mpls-generalized-cr-ldp-07.txt* (work in progress).

Ashwood-Smith, P. and Berger, L. [2002] "Generalized MPLS signaling — RSVP-TE extensions," in *Internet Draft*, *draft-ietf-mpls-generalized-rsvp-te-08.txt* (work in progress).

Awduche, D. O., Berger, L., Gain, D., Li, T, Swallow, G. and Srinivasan, V. [1999] "Extensions to RSVP for LSP tunnels," in *Internet Draft draft-ietf-mpls-rsvp-lsp-tunnel-04.txt*.

Awduche, D. Coltun, R., Rekhter, Y. and Drake, J. [2002] "Multiprotocol lambda switching: Combining MPLS Traffic Engineering with Optical Crossconnects," in *Internet Draft*, *draft-awduche-mpls-te-optical-04.txt* (work in progress).

Banerjee, A., Drake, J., Lang, J. P., Turner, B., Kompella, K. and Rekhter, Y. [2001] "Generalized multiprotocol label switching: An overview of routing and management enhacements," in *IEEE Communications Magazine*, pp. 144–150.

Banerjee, A., Drake, J., Lang, J. P., Turner, B., Kompella, K. and Rekhter. Y. [2001] "Generalized multiprotocol label switching: An overview of signaling enhancements and recovery techniques," in *IEEE Communications Magazine*, pp. 144–151.

Bertsekas, D. and Gallager, R. [1987] *Data Networks* (Prentice-Hall).

Chen, S. and Nahrstedt, K. [1998] *An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks*: *Problems and Solutions.*

Fratta, L., Gerla, M. and Kleinrock, L. [1973] "The flow deviation method: An approach to store-and-forward network design," in *Networks 3*, pp. 97–133.

Guerin, R., Ahmadi, H. and Naghshineh, M. [1991] "Equivalent capacity and its application to bandwidth allocation in high speed networks," in *IEEE Journal on Selected Areas in Communications*, pp. 968–981.

Guerin, R., Williams, D. and Orda, A. [1997] "QoS routing mechanisms and OSPF extensions," in *Proceedings of Globecom.*

Kodialam, M. S. and Lakshman, T. V. [2000] "Minimum interference routing with applications to MPLS traffic engineering," in *Proceedings of INFOCOM (2)*, pp. 884–893.

Kompella, K. and Rekhter, Y. [March 2001] "IS-IS extensions in support of generalized MPLS," in *Internet Draft*, *draft-ietf-gmpls-isis-extensions-02.txt* (work in progress).

Kompella, K. and Rekhter, Y. [November 2001] "OSPF extensions in support of generalized MPLS," in *Internet Draft*, *draft-kompella-ospf-gmpls-extensions-03.txt* (work in progress).

Rosen, E., Viswanathan, A. and Callon, R. [2001] "Multiprotocol label switching architecture," in *RFC 3031.*

Schormans, J. A., Pitts, J., Williams, K. and Cuthbert, L. [1994] "Equivalent capacity for on/off sources in ATM," in *Electronic Letters* **30**(21), 1740–1741.