A QoS Routing Mechanism for Reducing the Routing Inaccuracy Effects*

Xavier Masip-Bruin, Sergio Sánchez-López,
Josep Solé-Pareta, Jordi Dominigo-Pascual

Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya
C/Jordi Girona, 1-3 – 08034 Barcelona, Spain
{xmasip, sergio, pareta, jordid} @ac.upc.es

Abstract. In highly dynamic large IP/MPLS networks, when routing information includes not only topology information but also information to provide QoS, such as available link bandwidth, network state databases must be frequently updated. This updating process generates an important signaling overhead. Reducing this overhead implies having inaccurate routing information, which may cause both non-optimal path selection and a call-blocking increase. In order to avoid both effects in this paper we suggest a new QoS explicit routing mechanism called *BYPASS Based Routing* (BBR), which is based on bypassing those links along the selected path that potentially cannot cope with the traffic requirements. Routing algorithms derived from the proposed BBR mechanism reduce the call-blocking ratio without increasing the amount of routing control information.

1 Introduction

Emerging real time applications, such as video on demand, videoconferences or virtual reality, cannot be supported under the Internet conventional best effort model, due to both the variable delays in the queuing process and the problem of congestion. Before these applications can be deployed, the network has to be modified to support end-to-end QoS. *Multiprotocol Label Switching* (MPLS) [1] provides a fast-forwarding mechanism to route the traffic associated with each ingress-egress node pair by using labels, and can support QoS requirements if necessary. Nodes in an MPLS domain are named *Label Switching Routers* (LSR) and the path between an ingress-egress node pair is called *Label Switched Path* (LSP). In order to establish LSPs, MPLS networks use a signaling mechanism managed by the *Label Distribution Protocol* (LDP) [2], which allows the allocation and distribution of labels.

In the IP/MPLS context, the routing algorithms are extremely important. Traditionally, in IP networks, routing (OSPF, IS-IS) is done only considering network topology information, which is just updated due to either link/node failure or restoration. For

* This work was partially funded by the MCyT (Spanish Ministry of Science and Technology) under contract FEDER-TIC2002-04531-C04-02, and the CIRIT (Catalan Research Council) under contract 2001-SGR00226.

M. Ajmone Marsan et al. (Eds.): QoS-IP 2003, LNCS 2601, pp. 90-102, 2003.

QoS provision the routing algorithm must take into account more parameters than those related with topology and connectivity (QoS Routing). In the QoS Routing algorithms, the QoS attributes must be considered when selecting the path. This path selection has to be done according to the information contained in the *Traffic Engineering Database* (TED). Some examples of recently proposed QoS routing algorithms are the *Widest-Shortest Path* (WSP) [3], the *Shortest-Widest Path,* (SWP) [4], the *Minimum Interference Routing Algorithm,* (MIRA) [5], the *Profile-Based Routing,* (PBR) [6], and the *Maximum Delay-Weighted Capacity Routing Algorithm*, (MDWCRA) [7].

All the routing algorithms mentioned above rely on the accuracy of the information contained in the TEDs to optimize the path selection. However, many situations exist that lead to a certain degree of inaccuracy in the TED information. This uncertainty has negative effects in the path selection process, such as the increase of the call-blocking ratio. In order to avoid these effects, in this paper we suggest a new QoS explicit routing mechanism called *BYPASS Based Routing* (BBR), which is based on bypassing those links along the selected path that potentially cannot cope with the traffic requirements. The routing algorithms derived from the BBR mechanism reduce the call-blocking ratio without increasing the signaling overhead.

The remainder of this paper is organized as follows. In Section 2, the problem addressed in this paper is identified. Then, Section 3 presents the review of the related work and Section 4 describes our proposal. After that, Section 5 includes some performance evaluation results obtained by simulation. Finally, Section 6 concludes the paper.

2 Origins of Routing Inaccuracy and Scope of This Paper

The routing inaccuracy directly depends on the procedure used to update the TED information. Two different aspects, namely the number of nodes and links composing the network and the frequency at which the TED has to be updated, mainly affect this procedure.

In large networks the number of nodes and links able to generate updating messages must be limited. Such a limitation, clearly obtained in a hierarchical structure, as for example the PNNI [8], introduces an aggregation process that implies a loss of information. In fact, this aggregation process reduces the network to a logical network made of logical links and logical nodes in such a way that information about physical links and nodes is often lost. In this way, a certain routing inaccuracy is introduced in the link state information.

Concerning the frequency at which updating messages are sent throughout the network, in highly dynamic networks where link state changes are frequently expected, it is impractical to keep the network state databases correctly updated [9] due to the non desirable signaling overhead introduced by the large number of signaling messages needed. Such a signaling overhead is usually reduced by applying a certain triggering policy, which specifies when the nodes have to send a message to inform the network about changes in one or more of the links directly connected to them. In fact, a trade-off exists between the need of having accurate routing information and the need of reducing the number of updating messages. Three different triggering policies are described in [10], namely, *Threshold based policy*, *Equal class based policy* and *Exponential class based policy*. The first one is based on a threshold value (*tv*). Let b^i_r be the last advertised residual bandwidth of the link *i*, where the residual capacity of a link is defined as the difference between the total amount of bandwidth that this link can support and the bandwidth that is currently in use and b^i_{real} the current real residual bandwidth of that link. Then an updating message is triggered when

$$\frac{\left| b^i_r - b^i_{real} \right|}{b^i_r} > tv \ . \quad \textbf{(1)}$$

The other two policies are based on a link partitioning, in such a way that the total link capacity is divided into several classes. Being *Bw* a fixed bandwidth value, the *Equal class based policy* establishes its classes according to (0, Bw), (Bw, 2Bw), (2Bw, 3Bw), etc., and the *Exponential class based policy* according to (0, Bw), (Bw, (f+1)Bw), ((f+1)Bw, (f^2+f+1)Bw), etc., where *f* is a constant value. Then, an updating message is triggered when the link capacity variation implies a change of class.

This paper deals with the effects (call-blocking ratio increase and non-optimal path selection) produced in the path selection process when considering bottleneck requirements (e.g. bandwidth), and when the routing process relies on partial or inaccurate network state information. If the information contained in the network state databases is not perfectly updated, the routing algorithm could select a path unable to support the path incoming request, given that one or more links of that path could actually have less available resources than the specified by TED and that were required to set up that path. In this way, the incoming path request will be rejected in the setup process producing an increase of the call-blocking ratio. In this paper the inaccuracy introduced by the triggering policies is considered. Unfortunately this inaccuracy can be only defined for those triggering policies whose behavior can be perfectly modeled. This excludes the timer-based triggering policies of our study.

In summary, in this paper a new QoS explicit routing mechanism to improve the network performance for traffic flows with bandwidth requirements in a highly dynamic on-demand IP/MPLS environment is suggested. A simple IP/MPLS domain is considered, but in a previous work [11] a solution for computing and signaling explicit routes when two IP/MPLS domains are interconnected via an ATM backbone has been presented.

3 Review of Related Work

Several recent works exist in the literature addressing the problem of having inaccurate routing information when selecting a path. Documents [12-14] deal with finding the path that maximizes the probability of supporting the incoming traffic requirements. Based on this idea, R.Guerin and A.Orda [12] present different proposals for reducing the routing inaccuracy effects depending on the QoS constraint required by the incoming traffic. On one hand, in order to solve the problem for flows with bandwidth requirements authors suggest applying a modified version of the standard *Most Reliable Path* (MRP). On the other hand, when the objective is to compute an end-to-end delay bound, authors present two different approaches to deal with this problem, i.e. the rate-based approach and the delay-based approach and different solutions are generated for each model. The first approach has the advantage that once the delay is mathematically represented, the end-to-end delay bound only depends on the available bandwidth on each link. The second approach has the disadvantage that tractable solutions can be only applied for relatively limited cases. Nevertheless, authors introduce a simplification based on splitting the end-to-end delay guarantee to a several minor problems that extends the cases where solutions can be applied. In [13], D.Lorenz and A.Orda try to solve the problem of selecting an optimal path that guarantees a bounded delay by searching for the path most likely to satisfy this QoS requirement, namely the *problem MP (Most Probable Path)*. As in the solution presented in [12] for the delay-based approach, the complexity of this problem is reduced after splitting the end-to-end delay constraint in several local end-to-end delay constraints. How this partition is done and the optimization of this partition is analyzed as the *Optimally Partitioned MP Problem (OP-MP)*. Solutions based on using programming dynamics methods are presented to address the *problem OP-MP*. Also searching for the most likely path in [14] G.Apostopoulos et al, propose a new routing mechanism named *Safety Based Routing* (SBR), to address the routing inaccuracy effects when computing explicit paths with bandwidth constraints. The SBR is based on computing the *safety* (S), a new link attribute that is incorporated to the path selection process, which represents the effects of the routing inaccuracy in the link state reliability. The SBR can only be implemented when the performance and characteristics of the triggering policies that generate the routing inaccuracy are well known. Two algorithms inferred from the SBR mechanism are proposed in the document, the *Shortest-Safest Path* and the *Safest-Shortest Path*. The first selects the shortest path among the path that minimize the *safety* parameter and the second algorithm selects the path that minimizes the *safety* value among the shortest paths. Obtained simulation results show a lower bandwidth-blocking ratio when the *Shortest-Safest Path* (SSP) is the routing algorithm in use.

Another work was presented by S.Chen and K.Nahrstedt in [15]. Although the routing mechanism deals with the NP-complete delay-constraint least-cost routing problem, authors ensures that it can be perfectly applied to the bandwidth-constrained least-cost routing as well. Unlike other mechanisms it is not based on computing the path able to support the traffic requirements with a larger probability but rather a new multipath distributed routing scheme named *Ticket based probing* (TBP) is implemented. The TBP defines the imprecise state model by defining which information must be stored in every node, and then sends routing messages named *probes*, from the source node to the destination node to find the low cost path that fulfills the delay requirements. Obtained simulation results show that the TBP achieves high success ratio and low-cost feasible path with minor overhead.

Finally, in [16] the problem of selecting the most likely path, named *Maximum Likely Path selection* (MLPS) is implemented in an analog computer and solved by using a Hopfield Neural Network routing algorithm. This method has an important cost on complexity that can be obviate due to the analog structure in use. However, authors pointed out that this method is useful when hierarchical routing is applied and as a consequence small networks exist.

The main difference between our proposal and the existing solutions is the routing behavior when, even applying any of these new algorithms a path that really cannot cope with the traffic bandwidth requirements is selected. This situation is managed differently in the routing mechanism proposed in this paper. In fact, unlike other algorithms that reject the incoming LSP, our solution tries to jump over those links that impede the end-to-end path establishment by using a different pre-computed path.

4 BYPASS Based Routing

In this Section we describe a new QoS routing mechanism, named *BYPASS-Based Routing* (BBR) aiming to reduce the routing inaccuracy effects, that is the increase of the call-blocking ratio and the non-optimal path selection, in an IP/MPLS scenario. The BBR mechanism is based on computing more than one feasible route to reach the destination. The BBR instructs the ingress node to compute both the working route and as many paths to bypass the links (named *bypass-paths*) that potentially cannot cope with the incoming traffic requirements. Nevertheless as we discuss below, only those paths that bypass links that truly lack enough available bandwidth to support the required bandwidth are set up.

Note that the idea of the BBR mechanism is derived from the protection switching for fast rerouting discussed in [17]. However, unlike the use of protection switching for fast rerouting, in our proposal both the working and the alternative path (*bypass-path*) are computed simultaneously but not set up, they are only set up when required. The key aspects of the BBR mechanism to decide which links should be bypassed and how the *bypass-paths* are computed are the following:

Obstruct-sensitive links: A new policy must be added in order to find those links (*obstruct-sensitive links*, *OSLs*) that could not support the traffic requirements associated with an incoming LSP demand. This policy should guarantee that whenever a path setup message sent along the explicit route reaches a link that has not enough residual bandwidth to support the required bandwidth, this link had been defined as *OSL*.

Working path selection: Using the BBR two different routing algorithms can be initially analyzed. These algorithms are obtained from the combination of the Dijkstra algorithm and the BBR mechanism. Therefore, two different strategies may be applied:

- The *Shortest-Obstruct-Sensitive Path* (SOSP), computing the shortest path among all the paths which have the minimum number of *obstruct-sensitive links.*
- The *Obstruct-Sensitive-Shortest Path* (OSSP), computing the path that minimizes the number of *obstruct-sensitive links* among all the shortest paths.

***Bypass-paths*, calculation and utilization:** Once the working path is selected the BBR computes the *bypass-paths* needed to bypass those links in the working path defined as *OSL*. When the working path and the *bypass-paths* are computed, the working path setup process starts. Thus, a signaling message is sent along the network following the explicit route included in the setup message. When a node detects that the link by which the traffic must be flown has not enough available bandwidth to support the required bandwidth, it sends the setup signaling message along the *bypass-path* that bypasses this link. Thus, the set of bypassed links must always be defined as *OSLs* so that a feasible *bypass-path* exists. Moreover, it is important to note that the *bypass-paths* nodes are included in the setup signaling message as well, i.e. *bypass-paths* are also explicitly routed. In order to minimize the setup message size, *bypass-paths* are removed from the setup message when passing the link that bypass.

4.1 Description of the BYPASS Based Routing

Let *G(N,L,B)* describe a defined network, where *N* is the set of nodes, *L* the set of links and *B* the capacity bandwidth of the links. Suppose that a set of source-destination pairs *(s,d)* exists, named *P*, and that all the LSP requests occur between elements of *P*. Let b_{req} be the bandwidth requested in an element $(s,d) \in P$.

Rule 1. Let $G_r(N_r,L_r,B_r)$ represent the last advertised residual graph, where N_r,L_r and B_r are respectively the remaining nodes, links and residual bandwidths at the time of path setup. Let L^{os} be the set of *OSLs* (l^{os}), where l^{os} are found depending on the triggering policy in use. Therefore,

- Threshold policy: Let b^i_r be the last advertised residual bandwidth for a link l_i. This link l_i is defined as *OSL,* l^{os}_i if

$$l_i = l^{os}_i \mid l^{os}_i \in L^{os} \Leftrightarrow b_{req} \in (b^i_r(1\text{-}tv), b^i_r(1+tv)] \,. \tag{2a}$$

- Exponential class policy: Let $B^i_{l\_j}$ and $B^i_{u\_j}$ be the minimum and the maximum bandwidth values allocated to class *j* for a link l_i. So, l_i. is an *OSL,* l^{os}_i if

$$l_i = l^{os}_i \mid l^{os}_i \in L^{os} \Leftrightarrow b_{req} \in (B^i_{l\_j}, B^i_{u\_j}] \,. \tag{2b}$$

Rule 2. Let L^{os} be the set of *OSLs*. Let i_j and e_j be the edge nodes of a link $l^{os}_j \in L^{os}$. Let l_k be one link adjacent to l^{os}_j. The edge nodes of the *bypass-paths* to be computed are

$$(i_j, e_j) \qquad \Leftrightarrow \quad l_k \notin L^{os} \tag{3a}$$

or

$$(i_j, e_k) \text{ and } (i_k, e_k) \qquad \Leftrightarrow \quad l_k = l^{os}_k \in L^{os} \,. \tag{3b}$$

In this way two or more adjacent *OSLs* could be bypassed by a single *bypass-path*.

In accordance with these rules, in Fig. 1 a brief description of the BBR mechanism is presented. Steps 4 and 5 should be in detail explained. Once a link is defined as *OSL*, the BBR mechanism computes the *bypass-path* that bypasses this link. The *bypass-paths* are computed according to de *SOSP* performance, namely, the shortest path among those paths minimizing the number of *OSLs* is chosen. Other criteria could be used to select the *bypass-paths*, such as simply apply the *OSSP* performance or to maximize the residual available bandwidth. These different approaches are left for further studies.

BYPASS BASED ROUTING (BBR)

Input: The input graph $G_r(N_r,L_r,B_r)$. The *LSP* request is between a source-destination pair *(s,d)* and the bandwidth requirement is b_{req}.

Output: A route from *s* to *d* with enough *bypass-paths* to bypass the routing inaccuracy effects in the *obstruct-sensitive links*.

Algorithm:
1. Mark those links that are defined as *obstruct-sensitive link (OSL)* according to Rule 1.
2. Depend on the algorithm to be used, *OSSP*, *SOSP:*
   SOSP (shortest-obstruct-sensitive path):
   - Compute the weight of a link *l* as
     $$w(l) = 1 \Leftrightarrow l \in L^{os}, \qquad w(l) = 0 \Leftrightarrow l \notin L^{os}$$
   - Apply Dijkstra's algorithm to select the paths $p \in P$ that minimize the number of *OSLs* by using *w(l)* as the cost of each link
   - If more than one path *p* exists selects the path that minimizes the number of hops

   OSSP (obstruct-sensitive-shortest path):
   - Apply Dijkstra's algorithm to select the paths $p \in P$ that minimize the number of hops by using *w(l)* =1 as the cost of each link.
   - If more than one path *p* exists selects the path that minimizes the *OSLs*.
3. Compute a *bypass-path* for all the *OSLs* included in the selected path according to Rule 2.
4. Decide which *bypass- paths* must be used in accordance with real available resources in the path setup time.
5. Route the traffic from *s* to *d* along the setup path.

Fig. 1. BYPASS Based Routing Mechanism

Regarding the BBR complexity, two main contributors exist. On one hand selecting the shortest path by using a binary-heap implementation of the Dijkstra algorithm, introduces a cost of $O(L \cdot \log N)$. On the other hand, additional cost is introduced due to the *bypass-path* computation. Assuming that the *bypass-path* cannot include a network element which is also included in the working path, G(V, E) stands for the reduced network, where $V < N$ and $E < L$. Hence, a factor of $O(E \cdot \log V)$is added in order to compute one *bypass-path*. However, since a variable number M of *bypass-paths* may be computed along a working path, the cost is $O(M(E \cdot \log V))$. Being $\hat{M}$ an upper bound of the number of computed *bypass-paths* along a working path, the complexity reduces to $O(\hat{M}(E \cdot \log V))$, i.e., effectively to $O(E \cdot \log V)$. So, the complexity is $O(L \cdot \log N)+O(E \cdot \log V)$. This expression may be finally reduced if considering that the *bypass-paths* are computed based on a reduced graph. Therefore, the complexity is $O(L \cdot \log N)$.

4.2 Example for Illustrating the BBR Behavior

Before analyzing the suggested algorithms in a large topology, we can test the BBR performance in the simple topology shown in Fig. 2, which shows the residual network topology where the number associated to each link shows the residual available units of bandwidth.

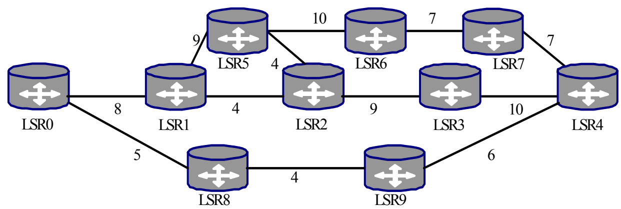

Fig. 2. Network topology used in the example

We suppose an *Exponential class* triggering policy with $f = 2$ and $Bw = 1$ (as used in [14]), in such a way that the resulting set of classes on each link are the following, (0,1], (1,3], (3,7], (7,15], etc. Moreover, we assume that an LSP incoming request is demanding b_{req} of 4 units of bandwidth between LSR0 and LSR4. In order to compare the BBR mechanism with other related work, the *Shortest-Safest Routing* algorithm presented in [14] is chosen as a sample of routing algorithm, which computes the path based on maximizing the "*probability of success*" of supporting the bandwidth requirements. Thus, the algorithms tested in this example are the SSP, WSP, SOSP, OSSP and the shortest path algorithm (SP) implemented in the OSPF routing protocol as a routing algorithm that does not consider the routing inaccuracy when selecting the path. Table 1 describes the link QoS parameters used to compute the path, where *B*, *Class* and *S* are the bandwidth, class and *safety* associated with each link. The *S* value has been computed according to the expressions found in [14]. Remind that *S* represents the probability that the requested amount of bandwidth is indeed available on a given link. Using this information the *BBR* mechanism is applied.

Table 2 shows different possible routes from LSR0 to LSR4 including the number of hops *H*, the number of *obstruct-sensitive links OSL*, the minimum last advertised residual bandwidth b_r^{min}, and the *safety* parameter *S*. As a result, different paths are selected depending on the algorithm in use, as it is shown in Table 2. Although the SOSP and the SSP algorithms select the same route, the key difference between both algorithms is the use of the *bypass-paths* when it is needed.

Table 1. Link QoS attributes

Link	B_t	*Class*	*S*
0-1	8	7,15	1
1-2	4	3,7	0,75
2-3	9	7,15	1
3-4	10	7,15	1

Link	B_t	*Class*	*S*
1-5	9	7,15	1
5-2	4	3,7	0,75
5-6	10	7,15	1
6-7	7	3,7	0,75

Link	B_t	*Class*	*S*
7-4	7	3,7	0,75
0-8	5	3,7	0,75
8-9	4	3,7	0,75
9-4	6	3,7	0,75

Table 2. Feasible routes and selected paths depending on the algorithm in use

Id	*Route (LSR)*	*H*	*OSL*	b_r^{min}	*S*
a	0-1-2-3-4	5	1	4	0.75
b	0-1-5-6-7-4	6	2	7	0.56
c	0-1-5-2-3-4	6	1	4	0.56
d	0-8-9-4	4	3	4	0.42

Alg	*Path*
SP	d
WSP	b
OSSP	d
SOSP	a
SSP	a

Once feasible routes have been computed, the *bypass-paths* selection process starts. If the SOSP algorithm is in use there is only one *OSL* in the route *a*, which can be bypassed by the path LSR1, LSR5 and LSR2. However, when the OSSP algorithm is in use, the process is much more complex since there are some *OSLs* that cannot be bypassed, e.g. link LSR8-LSR9. In this case the BBR cannot be applied. How to bypass *OSLs* that have not a *bypass-path* between its edges is a topic for further study. In this paper, as it has been pointed out above, the *bypass-paths* are always computed by minimizing the number of *OSLs*.

Finally, after computing the *bypass-paths*, a path setup message is sent along the working path. Each node checks the real available link bandwidth and depending on this value the setup message is sent through either the working or the *bypass-path*.

5 Performance Evaluation

In this section we compare by simulation the BBR algorithms introduced in this paper that is the SOSP and the OSSP algorithms, with the WSP and the SSP algorithms. We exclude the SWP due to its worse performance behavior shown in [18].

5.1 Performance Metrics

The parameters used to measure the algorithms behavior are the routing inaccuracy and the blocking ratio.

Routing Inaccuracy: This parameter represents the percentage of paths that have been incorrectly selected. It is defined as

$$\text{routing inaccuracy} = \frac{\text{number of paths incorrectly selected}}{\text{total number of requested paths}} . \quad (4)$$

A path can be incorrectly selected because of two factors. The first factor is the LSP request rejection when really there was a route with enough resources to support that demand. The second factor is the blocking of an LSP that initially was routed by the ingress node but, due to the insufficient bandwidth in an intermediate link, it is rejected.

Blocking Ratio: We use the bandwidth-blocking ratio defined as

$$\text{bandwidth blocking ratio} = \frac{\sum_{i \in rej\_LSP} \text{bandwidth}_i}{\sum_{i \in tot\_LSP} \text{bandwidth}_i} \quad (5)$$

where *rej_LSP* are the set of blocked demands and *tot_LSP* are the set of total requested LSP.

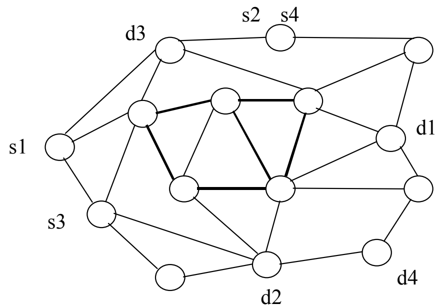

Fig. 3. Network topology used in simulations

5.2 Simulation Scenario

The simulations are performed over the network topology shown in Fig. 3, borrowed from [5], using the ns2 simulator extended with MPLS and BBR features. We use two link capacities, 622 Mb/s represented by a light line and 2.5 Gb/s represented by a dark line. The source nodes (s) and the destination nodes (d) are those shown in Fig. 3. Every simulation requests 1300 LSPs from s_i to d_i, which arrive following a Poisson distribution where the requested bandwidth is uniformly distributed between 1 Mb/s and 5 Mb/s. The holding time is randomly distributed with a mean of 60 sec. The *Threshold based triggering policy* and the *Exponential class based triggering policy* with f = 2, are implemented in our simulator.

5.3 Results

The results presented in this paper have been obtained after repeating the experiment 10 times, considering that every simulation lasts 259 sec. Fig. 4 shows the bandwidth-blocking ratio for the *Threshold* and the *Exponential class triggering* policies. The algorithms derived from the BBR mechanism (OSSP and SOSP) perform better than the WSP. In addition, while the OSSP presents similar results than the SSP, the SOSP substantially improves the SSP performance. Specifically, for the SOSP algorithm the Threshold value can be increased a 10% keeping the same bandwidth blocking ratio than the SSP.

Fig.5 represents the routing inaccuracy for both triggering policies. The SOSP algorithm presents the best behavior as well, that is, the SOSP is the algorithm that computes a lower number of incorrect routes.

Fig. 6 shows the cost of the BBR mechanism in terms of number of computed *by-pass-paths*. The figure shows that the cost is similar for both algorithms derived from the BBR mechanism. It reinforces the conclusion that the SOSP behaves better than the OSSP algorithm. The SSP and the WSP do not incur in the cost depicted in Fig. 5. Nevertheless, note that this cost is low given the benefits provided by the BBR mechanism shown in Fig. 4 and Fig. 5.

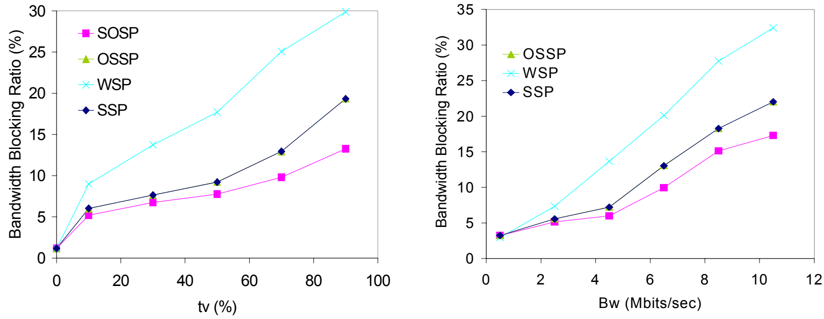

Fig. 4. Bandwidth Blocking Ratio for both Threshold and Exponential class triggering policies

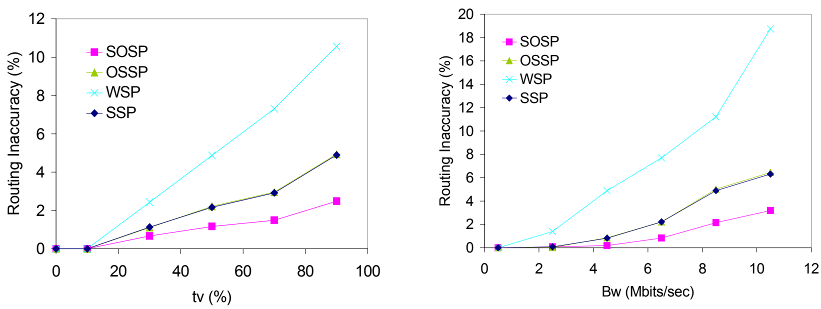

Fig. 5. Routing Inaccuracy for the Threshold and the Exponential class triggering policies

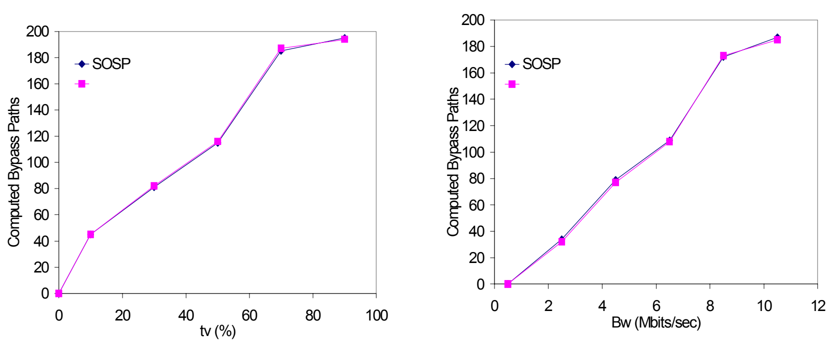

Fig. 6. Computed *bypass* paths for the Threshold and the Exponential class triggering policies

In summary, as a numeric example we take a *tv* value of 70 % and if we analyze the results provided by the BBR mechanism and those provided by the SSP algorithm, it is shown that the bandwidth blocking ratio presented by the SOSP (9.7 %) is substantially lower than that provided by the SSP (12.9 %). Regarding to the routing inaccuracy, the SOSP (1.49 %) presents a lower number of paths incorrectly selected when comparing to the SSP (2.91 %). In both cases the OSSP presents similar results than the SSP, and the WSP is that algorithm which presents the worst behavior. This is due to the fact that the WSP does not consider the routing inaccuracy when selecting the path. Finally, for a tv = 70 % the number of *bypass-paths* computed during the simulation for the BBR mechanism is close to 180 and almost the same for the SOSP and the OSSP algorithms. That means an overhead in computation of LSP about 14% but not in signaling as has been explained before.

6 Conclusions

In this paper a new QoS routing mechanism for establishing LSPs in an IP/MPLS network under inaccurate routing information has been suggested and its performance evaluated by simulation in comparison with the existing solution, the *Safety Based Routing* mechanism.

We called this new QoS routing mechanism, *BYPASS Based Routing* (BBR). The BBR minimizes the routing inaccuracy effects due to implementing a certain triggering policy to reduce the volume of updating messages. Basically, the main idea of BBR is to bypass those links that potentially are unable to support the traffic requirements associated with the incoming LSP request. These links are defined as *obstruct-sensitive links (OSL)* and a new mechanism is proposed to both define which links are to be *OSL* and find *bypass-paths* to bypass the *OSLs*.

Two algorithms are derived from combining BBR with the shortest path algorithm, namely the *Shortest-Obstruct-Sensitive* (SOSP) and the *Obstruct-Sensitive-Shortest* (OSSP). The simulation results obtained when comparing these BBR algorithms with the SSP and the WSP algorithms confirm the BBR effectiveness to improve the routing performance when the network state databases are not perfectly updated. In fact, the SOSP algorithm exhibits a lower bandwidth blocking ratio than the other tested routing algorithms, substantially improving the *Safety Based Routing* behavior. This improvement is achieved without incrementing the use of resources since the *bypass-paths* are established only when they are needed at the time of setup.

References

1. E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture", RFC3031, January 2001.
2. L.Anderson, P.Doolan, N.Feldman, A.Fredette, B.Thomas, "LDP Specification", RFC3036, Jan.2001.
3. R. Guerin, A. Orda, D. Williams, "QoS Routing Mechanisms and OSPF Extensions", In Proceedings of 2nd Global Internet Miniconference (joint with Globecom'97), Phoenix, AZ, November 1997
4. Z. Wang, J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications", IEEE JSAC, 14(7): 1288-1234, September 1996
5. M. Kodialam, T.V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering" In Proceedings of INFOCOM 2000.
6. S.Suri, M.Waldvogel, P.R.Warkhede, "Profile-Based Routing: A New Framework for MPLS Traffic Engineering".In Proceedings of QofIS 2001.
7. Y.Yang, J.K.Muppala, S.T.Chanson, "Quality of Service Routing Algorithms for Bandwidth-Delay Constrained Applications", in Proc. 9^{th} IEEEICNP2001, Riverside, Nov. 11-14, 2001.

8. Private Network-Network Interface Specification v1.0 (PNNI), ATM Forum , March 1996
9. G. Apostolopoulos, R. Guerin and S. Kamat, " Implementation and performance measurements of QoS Routing Extensions to OSPF", in Proceedings INFOCOM, March 1999, pp. 680-688
10. G. Apostolopoulos, R. Guerin, S. Kamat, S. Tripathi, "Quality of Service Based routing: A Performance Perspective" In Proceedings of SIGCOMM 1998.
11. S.Sánchez, X.Masip, J.Domingo, J.Solé, J.López "A Path Establishment Approach in an MPLS-ATM Environment", In Proceedings of IEEE Globecom 2001.
12. R.A. Guerin, A.Orda, "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms", IEEE/ACM Transactions on Networking, Vol.7, nº.3, pp. 350-364, June 1999
13. D.H.Lorenz, A.Orda, "QoS Routing in Networks with Uncertain Parameters", IEEE/ACM Transactions on Networking, Vol.6, nº.6, pp.768-778, December 1998.
14. G. Apostolopoulos, R. Guerin, S. Kamat, S.K. Tripathi, "Improving QoS Routing Performance Under Inaccurate Link State Information", In Proceedings of ITC'16, June 1999.
15. S.Chen, K.Nahrstedt, "Distributed QoS Routing with Imprecise Sdtate Information", in Proceedings of 7th IEEE International Conference of Computer, Communicati0ons and Nettworks, 1998
16. J.Levendovszky, A.Fancsali, C.Vegso, G.Retvari, "QoS Routing with Incomplete Information by Analog Computing Algorithms", in Proceedings QofIS, Coimbra, Portugal, September 2001.
17. T.M.Chen, T.H.Oh, "Reliable Services in MPLS", IEEE Communications Magazine, 1999 pp. 58-62
18. Q. Ma, P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees", In Proceedings of IEEE International Conference on Networks Protocols, October 1997.