Optimizing Routing Decisions Under Inaccurate Network State Information*

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

Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya
Avgda. Víctor Balaguer, s/n, 08800 Barcelona, Spain
{xmasip,sergio,pareta,jordid}@ac.upc.edu

Abstract. Maintaining accurate network state information in the Traffic Engineering Databases of each node along a network is extremely difficult. The BYPASS Based Routing (BBR) mechanism has appeared to reduce the effects produced in the network performance when selecting paths under inaccurate network state information. The BBR mechanism is based on applying the dynamic bypass concept. In this paper the BBR mechanism is modified to extend its applicability, and therefore to increase the benefits of its implementation.

Keywords: QoS routing, inaccurate routing decisions, routing scalability.

1 Introduction

Traditional IP networks are based on the best effort model to transport traffic flows between network clients. Since this model cannot properly support the requirements demanded by the emerging real time applications (such as video on demand, multimedia conferences or virtual reality), some modifications in the network structure, mainly oriented to optimize the network performance providing Quality of Service (QoS) guarantees are required. It is widely known that routing decisions strongly impacts on the QoS network behaviour. Having a powerful routing mechanism which takes into account the QoS parameters when selecting the routes (QoS Routing), targeting to an optimal end-to-end path selection is still a challenge.

Most QoS routing algorithms select paths based on the information contained in the network state databases (named Traffic Engineering Databases, TED, when including QoS parameters) stored in the network nodes. In fact, TED information is not only topology and connectivity but also QoS parameters are included. The management of such QoS parameters is done by the routing protocol, which must include a mechanism to collect, to distribute and to update all the parameters needed by the QoS routing algorithm. Important factors in the global routing behaviour are how and where routing decisions are taken. On the one hand, most QoS routing algorithms utilize the nominal link available bandwidth information to select paths. However, assuming that most clients generating network traffic do not completely use the assigned bandwidth, that is, do not strictly fulfil the Service Level Agreement (SLA), a

---

* This work was partially funded by the MCyT (Spanish Ministry of Science and Technology) under contract FEDER-TIC2002-04344-C02-02, the IST project E-Next under contract FP6-506869 and the CIRIT (Catalan Research Council) under contract 2001-SGR00226.

M. Ajmone Marsan et al. (Eds.): QoS-IP 2005, LNCS 3375, pp. 445–455, 2005.

difference exists between the nominal link utilization and the actual link utilization. This gap leads to non-efficient network resource utilization, since the path selection process is performed according to nominal link state information instead of actual link utilization information. This problem is addressed in [1], where authors propose a path selection algorithm named *Available Bandwidth Estimation Algorithm* based on performing an estimation of the future link utilization. This prediction is obtained by sampling the network state with a certain period of time which dynamically changes depending on the traffic characteristics and the conservatism requirements of the network domain.

On the other hand, being aware that most QoS routing algorithms take routing decisions in the source nodes (explicit routing) based on their global network state information, a mechanism to keep this network state information perfectly updated must be included in the routing protocol. This mechanism is mainly based on flooding updating messages to advertise network nodes about network state changes. In a large connection oriented packet switching network scenario, where changes are produced very often, this updating process generates a non-desirable signalling overhead. To reduce such a signalling overhead, a triggering policy is applied. In addition to the simple triggering policy based on sending updating messages at fixed intervals of time (*time based* triggers), usually three triggering policies are applied, namely *Threshold based policy*, *Equal class based policy* and *Exponential class based policy* [2]. The Threshold based policy lies in sending an updating message whenever the actual residual bandwidth differs (is lower or greater) from the last advertised residual bandwidth in a quantity defined by a threshold tv. The other two policies are based on a link bandwidth partitioning, in such a way that the total link capacity is divided into several classes. Being Bw (base class size) a fixed bandwidth value, the *Equal class based policy* establishes its classes according to $[(0, Bw), (Bw, 2Bw), (2Bw, 3Bw),\ldots]$, and the *Exponential class based policy* according to $[(0, Bw), (Bw, (f+1)Bw), ((f+1)Bw, (f^2+f+1)Bw),\ldots]$, where f is a constant value. Then an updating message is triggered when the link capacity variation implies a change of class.

Most QoS routing algorithms assume as a condition that the network state databases, from which the routing tables are built, represent a current picture of the network state. Therefore, routing algorithms select routes based on this hypothetical accurate network state information. Unfortunately, as a consequence of using triggering policies to reduce the signalling overhead produced by the updating process, maintaining accurate network state information cannot be always guaranteed. Thus, when the information contained in the network state databases is not perfectly updated, i.e. does not represent a current picture of the network state, the routing process might potentially select a path that is unable to support the call requirements. Consequently, a certain incoming connection that is initially allocated to a particular path would be rejected in the path set-up process. This is known as the routing inaccuracy problem.

This paper focuses on addressing the routing inaccuracy problem produced when the path selection process is performed under inaccurate nominal link state information because of the application of a certain triggering policy needed to reduce the signalling overhead produced by the updating process.

Several significant documents exist addressing the routing inaccuracy problem. Most relevant contributions were done by Guerin and Orda [3], Lorenz and Orda [4], T.Korkmaz and M.Krunz [5], Apostolopoulos et al. [6], Chen and Nahrstedt [7] and X.Masip et al. [8]. Table 1 shows the main characteristics of these contributions. A detailed review of these contributions can be also found in [8].

Table 1. Algorithms comparison

Ref.	*Based on*	*QoS Constr.*	*Main Goal*	*Routing*	*Routing information*
[3]	Probabil. appr.	Bw or delay	Reduce RIP* effects	Source	Global
[4]	Probabil. appr.	Delay	Reduce RIP* effects	Source	Global
[5]	Probabil. appr.	Bw/delay	Improve Computational efficiency	Source	Global
SBR [6]	Probabil. appr.	Bw	Reduce RIP* effects	Source	Global
TBP [7]	Multipath sche.	Delay	Reduce RIP* effects	Distributed	Local
BBR [8]	Dynam. bypass	Bw	Reduce RIP* effects	Source	Global

* RIP: Routing Inaccuracy Problem

One of the most recent contributions addressing the routing inaccuracy problem for bandwidth constrained applications is the BYPASS Based Routing (BBR) mechanism [8]. Although it is deeply demonstrated the benefits obtained in the global network performance (in terms of blocking probability) when using any routing algorithm inferred from the BBR mechanism, it is also noticed that the BBR mechanism suffers from two significant weaknesses, that is, inefficient *bypass-paths* utilization and large computational cost, both limiting the potential benefits of its implementation. It is the main goal of this paper to analyze to what extent such weaknesses degrade the global network performance as well as to provide new solutions to reduce such a performance degradation.

The rest of this paper is organized as follows. Section 2 shortly describes the BBR mechanism and its current existing weaknesses. Then, Section 3 provides the BBR with a solution to address them which are evaluated in Section 4. Finally, Section 5 concludes the paper.

2 BYPASS Based Routing Review

The BYPASS Based Routing (BBR) mechanism was proposed to reduce the effects produced by the routing inaccuracy problem in the global network performance for bandwidth constrained applications. All the algorithms inferred from the BBR mechanism are based on the dynamic bypass concept. By implementing such a concept some intermediate nodes along the selected path might reroute the setup message through alternative pre-computed paths, named *bypass-paths*, whenever these intermediate nodes do not have enough resources to cope with the incoming traffic requirements.

2.1 Basic Guidelines in the BBR Understanding

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

$G_r(N_r,L_r,B_r)$ represent the last advertised residual graph, where N_r, L_r and B_r are the remaining nodes, links and residual bandwidths at the time of path setup respectively. The main steps in the BBR performance are:

Obstruct-Sensitive Links. A new parameter is introduced in the path selection process to represent the routing inaccuracy. This parameter is translated into a new class of link. In this way, an *Obstruct-Sensitive Link (OSL)* is a link that potentially would be unable to support the traffic requirements. Specifically, a particular link is defined as *OSL* for an incoming connection request depending on the triggering policy in use. So, let L^{os} be the set of *OSL's*. Let $l^{os} \in L^{os}$ be a link of the residual graph G_r. As stated before a link l_i is defined as *OSL* as follows:

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

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

- *Exponential class triggering policy*: Let B_{l_j} and B_{u_j} be the minimum and the maximum bandwidth values allocated to class *j*. A link l_i is defined as *OSL*, that is l^{os}_i if

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

Working Path Selection. The first step implemented by any BBR routing algorithm is to mark those links to be defined as *OSL*. Concerning the path selection process, there are four routing algorithms inferred from the BBR mechanism so far, which are classified in two main groups, depending on the parameters used to select the paths, that is, the number of hops [8] and the last advertised residual bandwidth [9]. On the first group, the *SOSP* (Shortest-Obstruct-Sensitive Path) selects the shortest path among the paths with the minimum number of *OSLs*, and the *OSSP* (Obstruct-Sensitive-Shortest Path) selects the path with the minimum number of *OSLs* among the shortest paths. Then, on the second group the *WSOSP* (Widest-Shortest-Obstruct-Sensitive Path) selects the widest path among those selected by the SOSP (when there is more than one) and the *BOSP* (*Balanced-Obstruct–Sensitive Path*) selects that path that minimizes the F_p value among those paths that minimize the number of *OSLs*, where F_p, represents the relation between the maximum residual bandwidth and the number of hops along a path *p*, according to

$$F_p = n\left[\max\left(\frac{1}{b^i_r}\right)\right] \qquad i=1..n \tag{3}$$

where *n* is the number of hops and b^i_r is the available residual bandwidth on the link *i* in the path *p*. In this way, by using F_p as the cost of each link, the network load and the network occupancy is balanced in the path selection process. Simulation results obtained in [9] show that the *BOSP* algorithm is the best BBR algorithm in terms of blocking probability.

BYPASS Paths Computation. Once the route is selected, the BBR mechanism computes an alternative path that bypasses each *OSL*. These new paths are named *bypass-paths* and are used when an *OSL* cannot cope with the traffic requirements. If more than one *bypass-path* exists, the route that minimizes the number of *OSLs* is chosen (other options, such as either to minimize the number of hops or to maximize the residual available bandwidth, are left for further studies).

BYPASS Paths Usage. Being aware that the routing information updating process is not instantaneous, links that cannot cope with the traffic requirements are only known at the time of path setup. In this way, when a node detects a link *i* with $b^i_r<b_{req}$ it sends the setup message along the *bypass-path* which bypasses this link. Moreover, it is important to note that the *bypass-path* nodes are included in the setup signalling 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 the links that they are intended to bypass have been traversed. An analysis of the complexity of the BBR mechanism can be found in [8].

2.2 BBR Current Weaknesses

There are two main weaknesses in the current BBR implementation that significantly lead the BBR to a non-optimized performance. The former appears when analyzing the *bypass-paths* utilization. In fact, there are some network scenarios where the BBR mechanism cannot be applied because a *bypass-path* to bypass the link defined as *OSL* cannot be found. This is mainly motivated by the currently hard and close *bypass-path* definition. According to such a definition, the edge nodes of the *bypass-path* to be computed must be the same that the edge nodes of the bypassing *OSL* link. Therefore, whenever a disjoint route between the edges nodes of the link defined as *OSL* cannot be found, the BBR mechanism cannot be applied. In this case, the setup message will be blocked when the selected route lacks enough bandwidth (instead of rerouting the setup message along the *bypass-path* that should have been computed), driving to an inefficient bypass-path utilization. The latter appears when analyzing the cost associated with the BBR mechanism. It is worth noting that if a *bypass-path* must be computed for each link defined as *OSL*, the routing mechanism involves a large computational cost. A straightforward solution to this problem is to reduce the number of computed *bypass-path*. Unfortunately this means either there are links defined as *OSL* for which a *bypass-path* is not computed or there are links that despite fulfilling the requirements to be defined as *OSL* they are not. Consequently a more elaborated solution to improve the BBR performance must be sought.

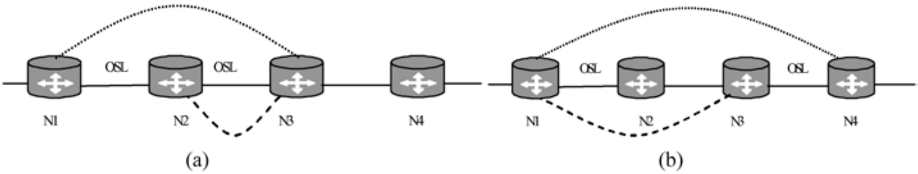

Fig. 1. BDP process

3 Optimizing the BYPASS Based Routing Mechanism

It has been shown in the last Section that the BBR mechanism suffers from two weaknesses, namely the inefficiency in the *bypass-path* utilization and the large computational cost. In this Section we show the impact of such weaknesses on the global network performance as well as we face them by extending the BBR mechanism with two new solutions. Thus, while the *BYPASS Discovery Process* (*BDP*) is proposed to address the inefficient *bypass-path* utilization, the *Network Load Dependent Policy* (*NLDP*) is proposed to reduce the computational cost required by the basic BBR mechanism.

3.1 BYPASS Discovery Process

The *BYPASS Discovery Process* (*BDP*) appears as a solution to extend the BBR applicability therefore increasing its benefits in the global network performance. In short, the BDP addresses the problem of inefficient *bypass-path* utilization by modifying the *bypass-paths* computation process targeting to increase the number of computed *bypass-paths*, so increasing the chances to apply the BBR mechanism.

Let i_j and e_j be the edge nodes of a link $l^{os}_j \in L^{os}$. Let e_{j+k} be the k node adjacent to e_j downstream along the working path. Then, the *BDP* computes *bypass-paths* in accordance with the following rules:

- Look for an alternative and disjoint route between the (i_j,e_j) pair (as done in the normal BBR mechanism).
- If there is not a feasible disjoint route between the (i_j,e_j) pair, then look for a route between the (i_j,e_{j+k}) pair, for $1 \leq k \leq d$, being e_{j+d} the destination node.

There are three main concerns to be deeply analyzed when applying the *BDP*. The first one states that feasible bypass routes cannot include any node belonging to the working path but the egress (destination) node. The two other concerns are analyzed by the examples drawn in Fig. 1. According to Fig. 1 (a) when two adjacent nodes are defined as *OSL* the *BDP* will compute a *bypass-path* to bypass both links and then another *bypass-path* to bypass the second link. The scenario is even more realistic in Fig. 1 (b) where there are two non-adjacent links defined as *OSL*. In such a scenario, despite the fact that the dash line could depict a tentative *bypass-path* to bypass link N1-N2 an optimal *bypass-path* would be that drawn by the dot line from N1 to N4 in order to reduce the number of used *bypass-paths*. Then, a *bypass-path* must also be computed to bypass link N3-N4 (not drawn in Fig. 2 (b)).

Fig.2 is used to make the *BDP* understanding easier. The performance of the *SOSP (Shortest-Obstruct-Sensitive Path)* and the *BOSP (Balanced-Obstruct-Sensitive Path)* algorithms when both include the *BDP* is analyzed. Suppose that updating messages are sent according to the *Exponential class triggering policy* with f=2 and Bw=1 (as used in [6]). Suppose that an incoming connection request arrives at N0 demanding b_{req} of 4 units of bandwidth to N4. Table 2 shows different tentative routes from N0 to N4 also including the parameters used by the *SOSP* and the *BOSP* algorithms to select the path. Moreover, it is also shown in detail in Table 2 the steps required by the BBR mechanism to compute paths, including the *BDP*, when applying both algorithms.

Table 2. BBR process

Id	*Route (N)*	*H*	*OSL*	b_r^{min}	F_p
a	0-1-2-3-4	4	1	4	1
b	0-1-5-6-7-4	5	2	7	0.71
c	0-1-5-2-3-4	5	1	6	0.83
d	0-8-9-4	3	3	4	0.75

BBR	*SOSP*	*BOSP*
1th step	Mark OSLs	Mark OSLs
2on step	a,c	a,c
3th step	a	c
4th step	1-2 (1,5,2)	5-2 (5,6,7,4)

According to the *SOSP* behaviour (Fig.2 (a)) a *bypass-path* exists (N1-N5-N2), represented by a dash line, to bypass the edges nodes of the link defined as *OSL,* that is N1-N2. However, when applying the *BOSP* algorithm, Fig.2 (b), a *bypass-path* to bypass the edge nodes of the link defined as *OSL*, that is N5-N2, cannot be found, so limiting the potential BBR benefits.

It is shown in the example that the BBR applicability depends on the routing algorithm in use. In fact, while using the *SOSP* does not involve any problem in the BBR applicability, the scenario is quite different when using the *BOSP*. In this case, if the *BDP* mechanism is not implemented, the required *bypass-path* cannot be computed. The *BDP* allows the BBR mechanism to compute a *bypass-path* (to bypass the *OSL*) from N5 to N4 (destination node), represented by a dash line, therefore extending and improving the BBR applicability.

It is worth noting that a collateral and negative effect because of including the *BDP* in the BBR mechanism is its impact on the computational cost, since as already defined, the more the number of computed bypass-paths the more the computational cost. Therefore, the solution also proposed in this paper to reduce the computational cost is even more relevant.

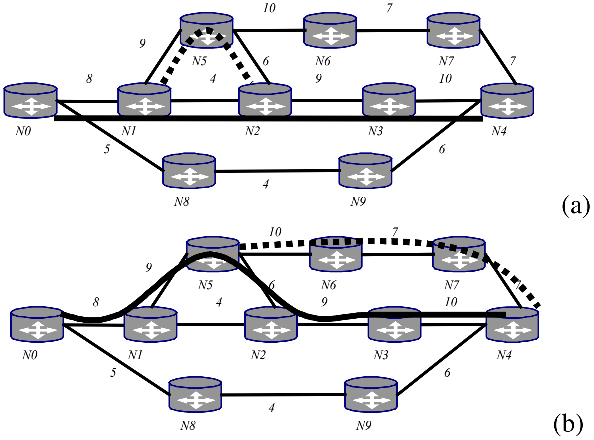

Fig. 2. Illustrative example

3.2 Network Load Dependent Policy

In [9] we carried out a simulation study concluding that the *BOSP* was the BBR routing algorithm showing the best performance in terms of blocking probability. Nevertheless, this algorithm exhibits large computational cost. In order to obtain both low blocking probability and low computational cost, we introduced a new policy named the *Network Load Dependent Policy* (*NLDP*) based on varying the number of *bypass-path* to be computed per route (*n_bp*) value according to the network load. The main *NLDP* concept is based on the following: when the network is not highly loaded no many links will be defined as *OSL*, so a low number of *bypass-paths* per route might be computed; instead, when the network is heavily loaded, a high number of *bypass-paths* per route will be needed to cover *OSL* definition. In order to limit the number of *bypass-paths* to be computed we decide ranging the *n_bp* value from 0 to 5.

4 Simulation Results

In this section we show by simulation the benefits of using both mechanisms, i.e. the *BDP* and the *NLDP*, when applying the BBR mechanism. In order to make the visibility easier we check both mechanisms separately.

The *NLDP* is evaluated over the NSFnet topology, by simulating 2500 incoming traffic connection following a Poisson distribution. Three different routing algorithms are evaluated to analyze the cost reduction together with the obtained bandwidth blocking reduction. Therefore, the *Widest Shortest Path* (*WSP*) [10] is selected as an example of QoS routing algorithm, the *Shortest-Safest Path* (*SSP*) [6] is selected as an example of QoS routing algorithm addressing the routing inaccuracy problem and finally the *BOSP* which has been defined as the BBR algorithm presenting larger benefits on bandwidth blocking reduction.

Table 3. Cost Analysis

n_bp	BW_{WSP}	BW_{SSP}	*Cost (LSP)*	*Cost (time)*
3	8.24%	4.62%	282	14.30%
Not limited	15.11%	7.99%	3178	161.23%
NLDP	7.62%	4.13%	102	5.17%

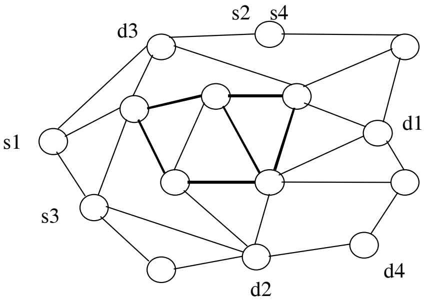

Fig. 3. Network topology used in simulations

Table 3 evaluates the impact on the BBR cost reduction because of applying the network load dependent policy. Three different situations are compared *n_bp* = 3, *n_bp* is not limited (a ∞ number of *bypass-paths* might be computed per route) and *n_bp* fulfils the *NLDP*. According to the *NLDP*, *n_bp* is in the range from 0 (network highly loaded) to 5 (network not loaded) *bypass-paths* per route. Results shown in Table 3 are obtained assuming *tv* = 80% (threshold value).

The values BW_{WSP} and BW_{SSP} stand for the difference in the bandwidth blocking obtained when comparing the *BOSP* to the *WSP* and the *SSP* respectively. The cost is represented in terms of both the total number of computed *bypass-paths* and the impact of computing these *bypass-paths* on the computational time. It is easy to note that the second scenario, where *n_bp* is not limited, is really not affordable because of the extreme cost. Analyzing the first and the last situation, while similar results in bandwidth blocking are obtained, the cost is significantly reduced when distributing *n_bp* proportionally to the network load.

Based on these results we can conclude that by using the *NLDP* the cost can be significantly reduced without substantially reducing the bandwidth blocking reduction.

Once we have shown the benefits obtained by including the *NLDP* in the BBR mechanism we show the benefits in the global network performance introduced when implementing the *BDP*.

In order to clearly identify the improvement obtained by the BBR mechanism when the *BDP* is also applied, the *BOSP* algorithm is evaluated in both situations, i.e., when the *BDP* is not implemented (named *BOSP*) and when the *BDP* is implemented (named *B/BDP*). Moreover, in order to evaluate the impact on the blocking probability because of the number of *bypass-paths* that can be computed per route, different simulations are carried out as a function of *n_bp*. The notation used in the figures to denote the *n_bp* values is *BOSP(n_bp)* and *B/BDP(n_bp).* The simulations are performed over the network topology shown in Fig. 3, using the ns2 simulator extended with BBR and *BDP* features. We use two link capacities, 622 Mb/s represented by a light line and 2.4 Gb/s represented by a dark line. Every simulation requests 1700 connection demands 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 120 seconds. The *Threshold* and the *Exponential class* (with $f = 2$) *triggering policies* are implemented in the simulations. The results presented in this paper have been obtained after repeating 300 seconds of simulation 10 times.

The parameters used to measure the algorithms behaviour are the routing inaccuracy and the bandwidth blocking ratio. The routing inaccuracy represents the total number of paths incorrectly selected among the total number of requested paths.

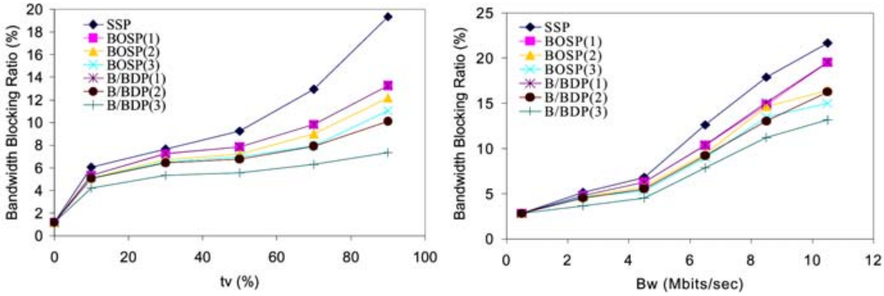

Fig. 4. Bandwidth blocking ratio for the *Threshold and the Exponential Class triggering policies*

The bandwidth blocking ratio for the *Threshold and Exponential class triggering policies* are depicted in Fig. 4. Focusing, for instance, in the *Threshold triggering policy* we can see that in any case a lower blocking is obtained when the BBR mechanism is applied compared to the *SSP* algorithm. Several conclusions can be obtained when deeply analyzing Fig. 4. Firstly, the lowest blocking is obtained when the *BDP* is applied, i.e., by the *B/BDP* algorithm. Secondly, although applying the *BDP* when *n_bp* = 1 hardly impacts on the blocking probability, we can conclude that the effects produced because of including the *BDP* in the BBR mechanism are completely dependent on the *n_bp* value. In fact, whereas a similar blocking is obtained for the

BOSP(1) and the *B/BDP(1)* algorithms, (13.27 % and 13.3 % respectively) a significant reduction of 2 % is obtained when *n_bp* = 2. Increasing the *n_bp* values leads to obtain an even more significant blocking reduction. Hence, the larger the number of computed *bypass-paths* per route the lower the blocking, that is, the blocking depends on the computational cost introduced in the network. Lastly, the larger the threshold value, i.e., the larger the inaccuracy, the larger the improvement obtained in the blocking reduction. In the worst conditions (the threshold *tv* of the triggering policy is 90 %), the bandwidth blocking ratio obtained due to applying the *BDP* process substantially improves that obtained by only applying the *BOSP*. For instance, when *n_bp* = 3, the obtained reduction in the blocking probability is 3.75 %.

Fig. 5 depicts the number of routes incorrectly selected for the *Threshold and the Exponential class triggering policies*. A similar behaviour is obtained for both triggering policies. As expected, the number of routes incorrectly selected decreases with the number of computed *bypass-paths*. Focusing again on the *Threshold triggering policy*, in the worst conditions (*tv* = 90 %), a reduction of 1.3 % is obtained when applying the *B/BDP(3)* algorithm compared to the *BOSP(1)* algorithm.

Finally, Fig. 6 shows the cost of applying the BBR mechanism with and without *BDP* capabilities. Moreover, it is also drawn the variation in the cost produced as a function of the *n_bp* value.

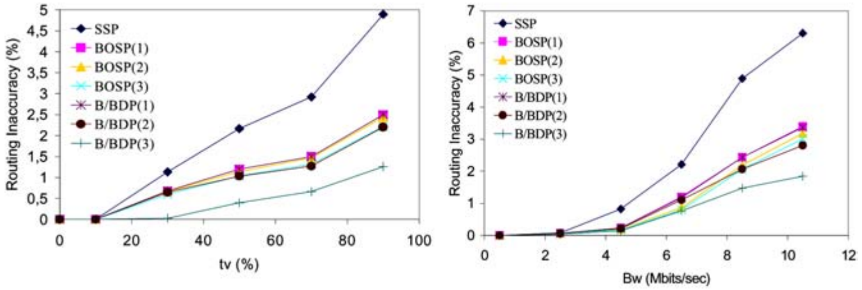

Fig. 5. Routing inaccuracy for the Threshold and the Exponential Class triggering policies

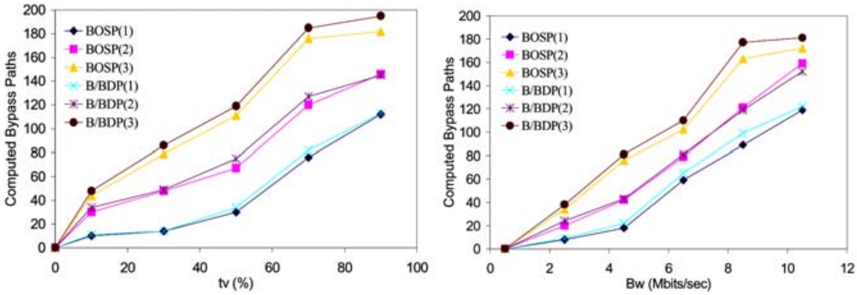

Fig. 6. Computed bypass-paths for the Threshold and the Exponential Class triggering policies

Definitely, we can conclude that extending the BBR with the *BDP* substantially improves the global network performance. It should be also noticed that a trade-off exists between the number of computed *bypass-paths* and the reduction obtained in the blocking probability.

5 Conclusions

This paper describes an optimization in the BBR mechanism to increase even more the benefits of applying this routing mechanism when selecting paths under inaccurate routing information. The optimization lies in on including two new mechanisms, the *BYPASS Discovery Process (BDP)* and the *Network Load Dependent Policy (NLDP)* to address two weaknesses of the BBR mechanism, that is, inefficient *bypass-path* utilization and a large computational cost. It is really important from the point of view of the BBR implementation to address such weaknesses since otherwise the potential benefits of applying the BBR mechanism are substantially reduced.

The *NLDP* appears to reduce the cost associated with the BBR mechanism. The main concept of this policy is based on dynamically limiting the number of *bypass-paths* to be computed according to the network load.

The *BDP* addresses the scenario where a *bypass-path* between the edge nodes of a link defined as *OSL* cannot be found. In this case, the *BDP* looks for a *bypass-path* between different edge nodes, so bypassing a set of links (instead of a single link) along the working path which must include the link defined as *OSL*. As a consequence the BBR can be correctly applied.

Both mechanisms are evaluated by simulation. Obtained results show a significant reduction in the blocking probability when including both mechanisms while keeping a reduced computational cost.

References

1. T.Anjali, C. Scoglio, J. de Oliveira, L.C. Chen, I.F. Akyldiz, J.A. Smith, G. Uhl, A. Sciuto, “A New path Selection Algorithm for MPLS networks Based on Available Bandwidth Estimation”, Proc. QofIS 2002, pp.205-214, Zurich, Switzerland, October 2002
2. G.Apostolopoulos, R.Guerin, S.Kamat, S.Tripathi, “Quality of Service based routing: a performance perspective”. Proc. SIGCOMM 1998
3. R.Guerin, A.Orda, D.Williams, “QoS routing mechanisms and OSPF extensions”, Proc. 2nd Global Internet Miniconference (joint with Globecom'97), Phoenix, AZ, November 1997
4. 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
5. T.Korkmaz, M. Krunz, “Bandwdith-Delay Constrained Path Selection Under Inaccurate State Information”, IEEE/ACM Transactions on Networking, vol.11, nº:3, pp.384-398, June 2003
6. G.Apostolopoulos, R.Guerin, S.Kamat, S.K.Tripathi, “Improving QoS routing performance under inaccurate link state information”, Proc. ITC’16, June 1999
7. S.Chen, K.Nahrstedt, “Distributed QoS routing with imprecise state information”, Proc.7th IEEE International Conference of Computer, Communications and Nettworks, 1998
8. X.Masip, S.Sánchez, J.Solé, J.Domingo, “A QoS routing mechanism for reducing the routing inaccuracy effects”, Proc. QoS-IP 2003, Milan, Italy, February 2003
9. X.Masip-Bruin, S.Sànchez-López, J.Solé-Pareta, J.Domingo-Pascual,, “QoS Routing Algorithms Under Inaccurate Routing Information for Bandwidth Constrained Applications”, Proc. IEEE International Communications Conference, ICC 2003, Anchorage, Alaska, May 2003.
10. R.Guerin, A.Orda, D.Williams, “QoS routing mechanisms and OSPF extensions”, Proc. 2nd Global Internet Miniconference (joint with Globecom'97), Phoenix, AZ, November 1997