Reducing the Effects of Routing Inaccuracy by Means of Prediction and an Innovative Link-State Cost

X. Masip-Bruin, E. Marín-Tordera, M. Yannuzzi, R. Serral-Gracià, and S. Sànchez-López

***Abstract*—The routing inaccuracy problem is one of the major issues impeding the evolution and deployment of Constraint-Based Routing (CBR) techniques. This paper proposes a promising CBR strategy that combines the strengths of prediction with an innovative link-state cost. The latter explicitly integrates a two-bit counter predictor, with a novel metric that stands for the degree of inaccuracy (seen by the source node) of the state information associated with the links along a path. In our routing model, Link-State Advertisements (LSAs) are only distributed upon topological changes in the network, i.e., the state and availability of network resources along a path are predicted from the source rather than updated through conventional LSAs. As a proof-of-concept, we apply our routing strategy in the context of circuit-switched networks. We show that our approach considerably reduces the impact of routing inaccuracy on the blocking probability, while eliminating the typical LSAs caused by the traffic dynamics in CBR protocols.**

***Index Terms*—Constraint-based routing, prediction routing, routing inaccuracy, update messages.**

I. Introduction

PROVIDING performance guarantees is a basic requirement to support current and future service demands. Constraint-based routing (CBR) is key in this regard, as being responsible for finding routes subject to performance constraints. Although many CBR strategies have been proposed in the literature, its adoption remains a challenging endeavor. In practice, several issues such as the computational complexity associated with CBR, the heterogeneity of current networks, and the trade-off between the overhead of frequently disseminating link-state information and the accuracy of this information, have so far hindered the deployment of CBR protocols.

In circuit-switched networks, the dissemination of link-state information in CBR protocols has a profound impact on both the performance and the scalability of the strategy used for finding and establishing the connections. Indeed, a significant part of the routing overhead is composed of messages used to keep link-state information on network nodes as updated as possible—notice that in large dynamic network scenarios a large number of update messages is required, clearly impacting on the CBR control plane overhead. Reducing this overhead will lead to having outdated link-state information, which will negatively impact on the performance of the network by significantly increasing the blocking of connection requests—usually referred to as the routing inaccuracy problem [1].

In this study, we present a novel CBR strategy addressing the routing inaccuracy problem. Our focus is on bandwidth constrained applications in the context of IP/MPLS networks. We shall show that our proposal considerably reduces the effects of the routing inaccuracy, while limiting the number of link-state messages to the minimum, i.e., to the distribution of topological changes in the network. This is achieved by including two concepts in the path computation; namely, prediction and an innovative cost function exploiting a link attribute that we refer to here as the link vulnerability (cf. Definition 1).

II. Related Work

Most of the existing contributions to the routing inaccuracy problem address the subject depending on the QoS constraint, either for delay constrained applications [2], for bandwidth constrained applications [3], [4], or for both constraints [1], [5]. With a different approach, [6] seeks to eliminate the inaccuracy implementing a QoS routing scheme based on a centralized server. On the other hand, whereas [7] focuses on localized QoS routing schemes, the authors in [8] focus on avoiding the use of TE (Traffic Engineering) information—a comprehensive list of works in the area along with an in-depth description of the abovementioned proposals can be found in [9]. Since our focus is on bandwidth constrained applications, we analyze in more detail the contributions in [3] and [4].

In [3], Apostopolulos et al. introduced the Shortest-Safest Path (SSP) algorithm. SSP computes paths according to a safety parameter, which provides a measure of the probability that the total bandwidth required (b_{req}) is indeed available in the sequence of links composing each path.

In [4], the authors proposed the concept of Bypass-Based Routing (BBR), which is based on computing alternative routes for some intermediate links that might be unavoidable when setting the path—considering that the path has been selected according to outdated link-state information. Experimental results in [4] show that BBR outperforms the SSP algorithm presented in [3].

Our contribution differs from the above works in the fact that the CBR strategy proposed here is not only able to efficiently deal with the routing inaccuracy problem, but also guarantees that the Link-State Advertisements (LSAs) are restricted to the distribution of topological changes in the network—the availability of resources along a path is predicted rather than updated through conventional LSAs.

III. The Proposed Routing Strategy

The prediction that we use for deciding about the availability of a route is the one proposed in [10]. That work

Manuscript received November 25, 2009. The associate editor coordinating the review of this letter and approving it for publication was F. Theoleyre.

This work was supported in part by the Spanish Ministry of Science and Innovation under contract TEC2009-07041, and the Catalan Research Council (CIRIT) under contract 2009 SGR1508.

The authors are with the Advanced Network Architectures Lab (CRAAX), Technical University of Catalonia (UPC), 08800 Vilanova i la Geltrú, Spain (e-mail: {xmasip, eva, yannuzzi, rserral, sergio}@ac.upc.edu).

Digital Object Identifier 10.1109/LCOMM.2010.05.092312

1089-7798/10$25.00 © 2010 IEEE

proposes the Predictive-Selection of a Route (PSR) algorithm, which predicts the route availability taking into account the outcome (success or failure) of the establishment of previous connection requests along the same route. The history of previous outcomes for a route is recorded by means of two-bit counters. Thus, the two-bit counter values 0 and 1 stand for route availability while the values 2 and 3 stand for route unavailability. When a route is selected, if the connection can be set up the corresponding two-bit counter is decreased, otherwise, if the connection is blocked, the two-bit counter is increased. The strength of the PSR algorithm is that routes are computed entirely based on the source node prediction, eliminating the need for updating link-state metrics in the PSR routing protocol [10].

The vulnerability, on the other hand, is used to avoid selecting links whose available bandwidth is close to the requested bandwidth. This is because such links are prone to be unavailable during the connection establishment if routing information is not perfectly updated. More precisely, let $b_i^{(s)}$ be the available bandwidth on link i of a route computed by the source node s. This bandwidth is locally determined by the source node, according to the node's previous allocations of resources along a route using link i (recall that the available bandwidth b_i is never updated through LSA in our CBR strategy, so $b_i \leq b_i^{(s)}$). Let b_{req} denote the bandwidth requested by an incoming connection, and let ε, with $\varepsilon \in [0, 1)$, be a predefined threshold reflecting the degree of inaccuracy tolerated by the CBR protocol. Then, the link vulnerability is a Boolean function locally computed by the source node s defined as follows.

Definition 1: The vulnerability $v_i^{(s)}$ of link i is defined as:

$$v_i^{(s)} = \begin{cases} 1 \quad \textit{iff} \quad 1 - \dfrac{b_{req}}{b_i^{(s)}} < \varepsilon \\ 0 \quad \text{otherwise} \end{cases} \quad \text{with} \quad b_{req} \leq b_i^{(s)} \tag{1}$$

With the above rationale, we propose the *Balanced-Vulnerable-Predictive Path* (BVP^2) algorithm to compute routes based on combining prediction and the link vulnerability concepts.

A. The BVP^2 Algorithm

The BVP^2 algorithm is aimed at minimizing the impact of the routing inaccuracy in the blocking, as well as minimizing the number of link-state updates and balancing the network load. The first target is achieved by introducing the link vulnerability attribute $v_i^{(s)}$ in the path selection process. The second target is achieved by introducing prediction concepts, in a way that completely eliminates the need for resource usage LSAs in the CBR protocol. The third target is achieved by adding the available bandwidth in the path selection process. All these, are effectively combined in a link-state cost as follows.

Let $P_j^{(s)}$ be the two bit counter for route j implementing the predictor described in [10]. Let N_j denote the length of route j in terms of hops, and, $V_j^{(s)} = \sum_{i=1}^{N_j} v_i^{(s)}$, be the number of links along route j having their vulnerability attribute set to 1. Let $B_j^{(s)} = [b_1^{(s)}, \ldots, b_{N_j}^{(s)}]$ be the array of bandwidth availability on each link i of route j, according to the locally assigned bandwidths made by node s. The cost function is thus given by:

$$C_j^{(s)} = P_j^{(s)} + N_j V_j^{(s)} \max\left(\frac{1}{b_i^{(s)}}\right) \ \forall \ b_i^{(s)} \in B_j^{(s)} \tag{2}$$

In this cost function, $P_j^{(s)}$ has the main weight; the second term is an order of magnitude smaller. The decision is taken according to $P_j^{(s)}$, but for routes with equal $P_j^{(s)}$, the second term breaks the tie.

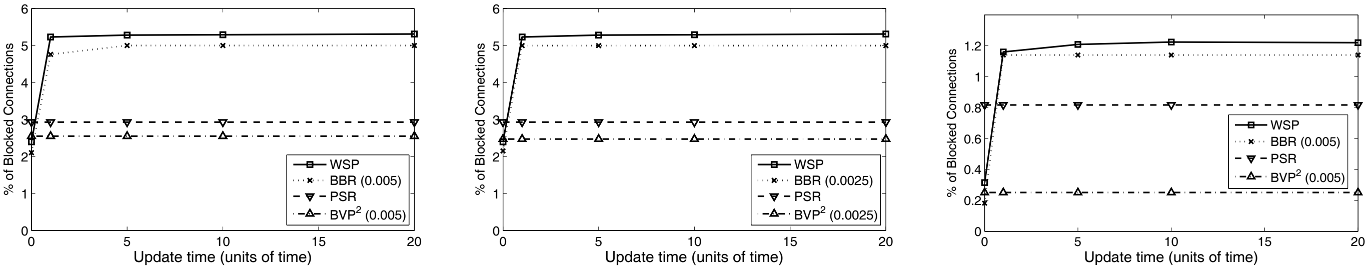
As shown in Algorithm 1 below, BVP^2 considers k precomputed paths between the source and destination, and selects the route minimizing the link-state cost in (2). The complexity of the algorithm is the following: from an offline phase the algorithm receives as input k shortest routes that are precomputed by means of Dijkstra, with a complexity of $O(k(|E| + |V|log|V|))$. For the online phase, when a connection request arrives, the source node performs the pseudo-code shown in Algorithm 1, which has a complexity of $O(1)$ when $k \ll \min\{|E|, |V|\}$, and the network diameter is $\text{diam}(G(V, E)) \ll \min\{|E|, |V|\}$.

IV. Performance Evaluation

The algorithm proposed in this paper has been evaluated through extensive simulations over the Pan European Network, with 28 nodes and 41 links. Connection arrivals, between all possible source-destination node pairs (full mesh) are assumed to be Poisson, and all the links have the same capacity (normalized bandwidth 100%). The arriving connections require a certain percentage of the total bandwidth, and all connection requests have average holding and inter-arrival

Algorithm 1 BVP^2

```
Require: $(b_{req}, k, \varepsilon, [(B_1, P_1), \ldots, (B_k, P_k)])$
Ensure: (Assigned_Route, $P_{\text{Assigned\_Route}}$)
  $C_{old} = \infty$
  Assigned_Route $= \varnothing$
  for $j = 1$ to $k$ do
    $b_{min} = \min(B_j)$, $B_j = [b_{1j}, \ldots, b_{|B_j|j}]$
    if $(b_{req} \leq b_{min})$ then
      $V_j =$ compute_vulnerability(route $j, B_j, b_{req}, \varepsilon$)
      $C_j = P_j + |B_j| V_j \left(\frac{1}{b_{min}}\right)$
      if $(C_j < C_{old})$ then
        Assigned_Route $= j$
        $C_{old} = C_j$
      end if
    end if
  end for
  if (Assigned_Route $\neq \varnothing$) then
    Decrease the two-bit counter $P_{\text{Assigned\_Route}}$
  else
    Increase the two-bit counter $P_{\text{Assigned\_Route}}$
  end if
```



(a) (b) (c)

Fig. 1. (a) Blocking probability for $\varepsilon = 0.005$ and average requested bandwidth of 1.25%; (b) blocking probability for $\varepsilon = 0.0025$ and average requested bandwidth of 1.25%; (c) Blocking probability for $\varepsilon = 0.005$ and average requested bandwidth of 1%.

times of 10 units, which means that at any moment 378 active connections may be active in the network. We simulate 37800 connection requests. We also consider $k = 4$ precomputed shortest routes. In order to analyze the effects of the traffic load on the algorithm, different simulations have been run with different values of the average bandwidth demanded by every incoming request between each source destination pair in terms of a percentage of the total capacity. The performance of BVP^2 is compared with PSR (only predicting concepts), BBR (considered the best algorithm according to [4]) and the Widest Shortest Path (WSP) [11], which does not consider the routing inaccuracy problem to compute routes. Two sets of simulations are shown keeping the ε threshold constant to 0.0025 and 0.005. In order to make the comparison possible, we also use the ε value in the BBR algorithm to select the links to be bypassed.

Figures 1a and 1b show the blocking probability obtained with BVP^2, BBR, WSP and PSR, for average bandwidth requests of 1.25% of the total capacity of any link of the network. The results show that, by construction, BVP^2 is independent of the update time interval, and it behaves better than the other evaluated algorithms. BVP^2 behaves better than PSR because the latter only selects one path (the first one meeting the bandwidth estimation and two-bit counter constraints). However, BVP^2 compares all k precomputed paths and takes decisions based on additional information (number of hops, traffic balance, and the link vulnerability attribute). Only for an update time equal to 0 (perfect knowledge) BBR and WSP perform better than PSR and BVP^2.

Figure 1c shows the results for an average bandwidth of every connection request of 1% and $\varepsilon = 0.005$. We observe that for this lower traffic load, the behavior is the same. For the sake of simplicity (and space) we do not present results for higher values of traffic, since all the routing protocols reach percentages of blocking around $15-20\%$, which is not realistic in practice.

Finally, Figure 2 depicts the blocking of BVP^2 as a function of ε, for 1.25% of traffic load. There are two values of ε with minimum percentage of blocked connections, $\varepsilon = 0.0125$ and $\varepsilon = 0.2$. Note that $\varepsilon = 0.0125$ corresponds to a threshold equal to the average requested bandwidth, 1.25%.

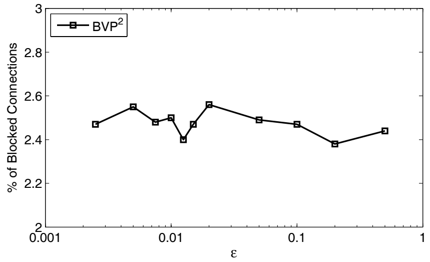


Fig. 2. Blocking probability versus ε for an average requested bandwidth of 1.25%.

V. Conclusions

The contribution of this paper is centered on the provision of a Constraint-based routing (CBR) algorithm that improves network performance, while eliminating the typical Link-State Advertisements (LSAs) caused by the traffic dynamics in CBR protocols. Further work would go to analyze the impact of the different blocking factors in equation (2).

References

[1] R. Guerin and A. Orda, "QoS routing in networks with inaccurate information: theory and algorithms," IEEE/ACM Trans. Networking, vol. 7, no. 3, June 1999.

[2] D. H. Lorenz and A. Orda, "QoS routing in networks with uncertain parameters," IEEE/ACM Trans. Networking, vol. 6, no. 6, Dec. 1998.

[3] G. Apostopoulos, *et al.*, "Improving QoS routing performance under inaccurate link state information," in *Proc. ITC'06*, June 1999.

[4] X. Masip-Bruin, *et al.*, "QoS routing algorithms under inaccurate routing information for bandwidth constrained applications," in *Proc. IEEE ICC 2003*, Alaska, May 2003.

[5] T. Korkmaz and M. Krunz, "Bandwidth-delay constrained path selection under inaccurate state information," IEEE/ACM Trans. Networking, vol. 11, no. 3, June 2003.

[6] S. Kim and M. Lee, "Server based QoS routing with implicit network state updates," in *Proc. IEEE Globecom*, San Antonio, TX, Nov. 2001.

[7] S. Nelakuditi, *et al.*, "Adaptive proportional routing: a localized QoS routing approach," in *Proc. IEEE INFOCOM*, Israel, Mar. 2000.

[8] T. Anjali, *et al.*, "A new path selection algorithm for MPLS networks based on available bandwidth estimation," in *Proc. QofIS 2002*, Oct. 2002.

[9] X. Masip-Bruin, *et al.*, "Routing bandwidth constrained applications under inaccurate network state information," technical report UPC-DAC-RR-CBA-2007-8. Available from: http://gsi.ac.upc.edu/reports/2007/39/report_OBBR_last_im.pdf.

[10] E. Marn-Tordera, *et al.*, "The prediction approach in QoS routing," in *Proc. IEEE ICC*, Istanbul, Turkey, May 2006.

[11] R. Guerin, *et al.*, "QoS routing mechanism and OSPF extensions," in *Proc. 2nd Global Internet Miniconference* (joint with GLOBECOM'97) 1997.