Dynamic Path Substitution Localized QoS Routing Scheme

Elmabrook B. Mustafa, Michael E. Woodward
University of Bradford, Bradford, BD7 1DP, UK
{E-mail: e.b.m.mustafa, m.e.woodward} @ bradford.ac.uk

Abstract — **Localized Quality of Service (QoS) routing schemes make routing decisions based on locally collected network state. But most QoS schemes proposed use global network state that requires periodic exchanges of network state among the routers. This results in communication overheads and processing overhead. The Localized (QoS) routing schemes avoid these problems by eliminating periodic exchange of QoS state information. We develop Dynamic Path Substitution Localized QoS Routing (DPSLR) and demonstrate through simulation that it outperforms the existing Credit Based Localized QoS Routing (CBR) which is the most efficient localized routing scheme to date.**

***Keywords* — QoS routing, Localized QoS routing.**

I. INTRODUCTION

Quality of Service (QoS) routing is concerned with the problem of how to select a path for network traffic flow such that the selected path assures the flow requirements. Traditional QoS routing schemes such as source routing schemes and distributed routing schemes make routing decisions based on global QoS state information. This requires periodic exchange of QoS state information using a link state algorithm among the routers which results in extra communication overheads in the network and processing overheads in the routers.

In source routing schemes [1, 2, 3], each source node must have global QoS state information of the network in order to make routing decisions. Nodes in source routing schemes use link state algorithm to advertise their state to other nodes in the network. This process of advertising the global state among nodes results in high communication overhead and processing overhead in routers.

Elmabrook B. Mustafa is PhD Candidate with the School of Informatics (Computer Department), University of Bradford, Bradford (e-mail: e.b.m.mustafa@bradford.ac.uk).

Prof. M. Woodward is with School of Informatics (Computer Department) , University of Bradford, Bradford (e-mail: M.E.Woodward@Bradford.ac.uk).

Distributed routing schemes [4, 5, 6] make their routing decisions based on global QoS state information of the network. Thus they share the abovementioned drawbacks of source routing.

Localized QoS routing schemes [7, 8, 9] attempt to avoid these problems by making the source node to make routing decisions based locally collected QoS state information. In localized routing schemes each node maintains a set of candidate paths to each possible destination in the network and routes network traffic along these paths. The selection mechanism of the candidate paths plays an important role in localized routing.

In this paper we introduce the dynamic path substitution localized QoS routing (DPSLR) scheme. This is based on CBR. DPSLR is an improved version of CBR by adding the feature of dynamically swapping loaded candidate paths with one which is not loaded. We compare its performance through extensive simulation with CBR proposed in [8, 9].

II. RELATED WORK

Localized QoS routing schemes have been proposed to overcome the problems of traditional QoS routing schemes (source QoS routing and distributed QoS routing).

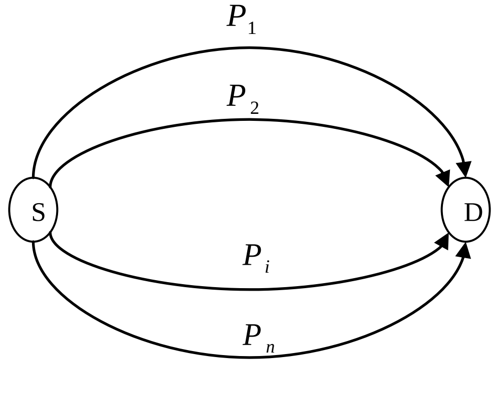


Figure1: Candidate paths

Localized QoS routing schemes make each source node (S) in the network maintains a set of candidate paths ($p_1, p_2, ... p_i ..., p_n$) with respect to each other node (D) in the network as shown in figure 1 and routes network traffic based on locally collected QoS state information. Therefore, localized routing schemes have several advantages as stated in [7]. The first advantage is the reduction of the communication overhead due to inferring the global state information exchange. The second, is the reduction of both processing time and memory in core routers. These result from inferring those router from keeping and updating the global network state database.

The most recent localized QoS routing scheme is CBR which will be used to compare the performance of our scheme. CBR maintains a set of candidate paths **R** for each node in the network and associates a crediting mechanism for each path in the candidate path set. The candidate path set **R** contains two types of candidate paths minhop paths $R^{\min}$ and alternate paths R^{alt}, where **R** = $R^{\min} \bigcup R^{alt}$. Every path **P** ∈ **R** is associated with credit. This credit will be credited on network traffic acceptance or debited on network traffic rejection. The CBR selects the path **P** ∈ **R** that has the largest credit and routes network traffic flow along that path. Detailed description of CBR can be found in [8][9].

III. THE PROPOSED ALGORITHM (DPSLR)

DPSLR is based on a simple idea that improves the performance of CBR. The proposed algorithm uses a simple mechanism based on path substitution. DPSLR like CBR maintains a set of candidate paths **R** which contains minhop paths $R^{\min}$ and alternate paths R^{alt}. In addition to that the DPSLR maintain a set of reserved paths R^{res}. Hence, the set of candidate paths R = $R^{\min} \bigcup R^{alt} \bigcup R^{res}$. The DPSLR keeps monitoring the credit of $R^{\min}$ and R^{alt}. And, it selects the path **P** ∈ $R^{\min} \bigcup R^{alt}$ that has the highest credit. Then, it routes the network traffic in the selected path. Also, the DPSLR algorithm keeps history of credits of every **P** ∈ **R** and uses this history to calculate the average of credits. This average will be used to govern the substitution of the path **P** ∈ $R^{\min} \bigcup R^{alt}$ with a path **P**∈ R^{res} when this average is less than MIN_CREDIT. For example, if a path $p_i \in R^{\min} \bigcup R^{alt}$ has credits ($c_1, c_2, c_3 ..... c_n$) then the average of these credits is:

$$\text{CREDIT\_AVE RAGE} = \frac{\sum_{i=1}^{n} c_i}{n}.$$

The DPSLR algorithm compares the CREDIT_AVERAGE with MIN_CREDIT and substituted it with a path **P**∈ R^{res} which has the highest credit.

IV. SIMULATION AND REDULTS

We developed a simulator using Java programming language. This simulator uses network topology that has been created by the Brite generator [10] using Waxman's Model. In all topologies links are assumed to be bidirectional and have the same capacity **C** in each direction (C=150MB). Network traffic arrives at each source node (selected randomly) according to a Poisson process with λ and the destination node is selected randomly. Network traffic duration is exponentially distributed with mean 1/μ and bandwidths are uniformly distributed within two intervals [0.1-2MB]. The offered network load is $\rho = \frac{\lambda N \bar{b} \bar{h}}{\mu L C}$ where N is the number of nodes, $\bar{b}$ is the average bandwidth required by network traffic $\bar{h}$ is the average path length and L is the number of links in the network.

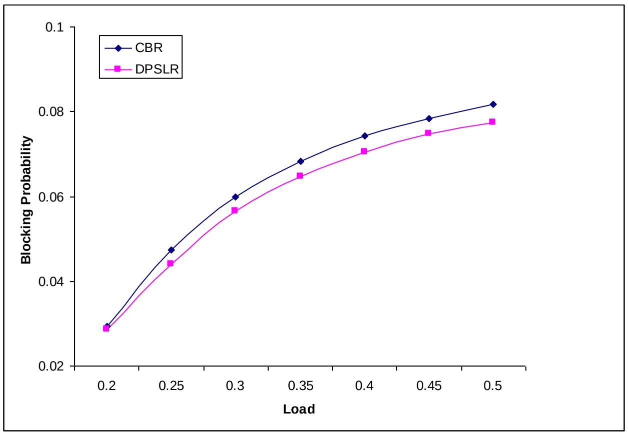
The performance of the routing algorithm may vary across different underlying network topologies and load conditions. Therefore our simulation considers different types of network topologies and different load conditions.

The parameters for CBR and DPSLR algorithms are MAX_CREDIT = 5, MIN_CREDIT=2.5 and Φ = 1. Blocking probabilities are calculated based on the last 10 connection requests. The candidate path sets $R^{\min}$, R^{alt} and R^{res} based on minhop and minhop+1. Each simulation run simulates the arrival of 2,000,000 connection requests. We use network traffic bandwidth blocking probability as the main performance metric, which is defines by:

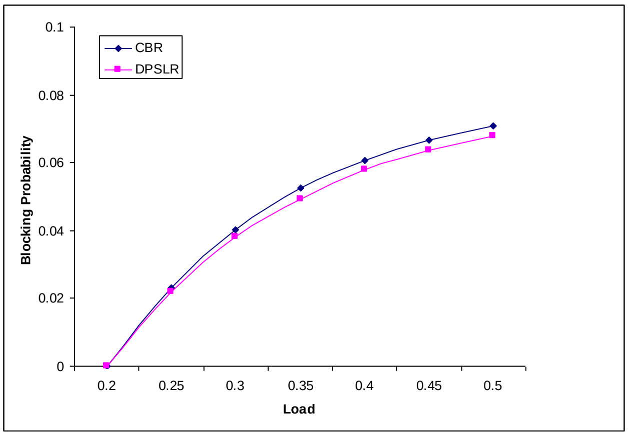
$$\text{Network traffic bandwidth BP} = \frac{\sum_{f \in B} bandwidth(f)}{\sum_{f \in C} bandwidth(f)}$$

Where BP is short for Blocking Probability, B is the set of blocked network traffic and C is the set of total requested bandwidth.

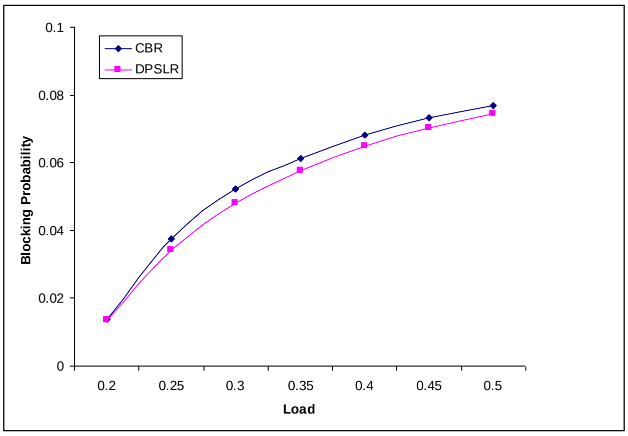
Figure 2, compares the performance of the two algorithms in terms of network traffic blocking probability under different network topologies and different load conditions. The various offered loads are plotted against the network traffic blocking probability using (a) RAND20, (b) ISP and (c) RAND40. From the figure we note that the DPSLR algorithm outperforms the CBR algorithm in all topologies as the network gets loaded.



(a) Random topology (RAND20)



(b) ISP topology



(c) Random topology (40 Nodes)

Figure 2. Impact of network topology and load

V. CONCLUSION

In this paper we present the DPSLR algorithm. It is based on a substitution mechanism that performs routing using a locally collected QoS state network. We compare its performance with the most efficient localized routing scheme (CBR) and we demonstrate through simulation that our algorithm outperform CBR.

REFERENCES

[1] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin,A. Orda and T. Przygienda., QoS Routing Mechanism and OSPF Extensions, RFC 2676, 1999.

[2] S. Chen and K. Nahrstedt, An Overview of Quality-of-Service Routing for the Next Generation HighSpeed Networks: Problems and Solutions, IEEE Network Magazine, 12 (1998), pp. 64 -79.

[3] R. Guerin and A. Orda, QoS-Based Routing in Networks with Inaccurate Information: Theory and Algorithms, IEEE/ACM Transactions on Networking, 7 (1999), pp. 605-614.

[4] S. Chen and K. Nahrstedt, Distributed QoS Routing with Imprecise State Information, 7th IEEE International Conference on Computer, Communications and Networks, IEEE, Lafayette, LA, 1998, pp. 614-621.

[5] S. Chen and K. Nahrstedt, Distributed Quality-of-Service routing in HighSpeed Networks Based on Selective Probing, 23rd Annual Conference on Local Area Networks (LCN'98), IEEE, 1998, pp. 80--89.

[6] S. K. Kweon and K. G. Shin, Distributed QoS routing using bounded flooding, University of Michigan, 1999.

[7] S. Nelakuditi, Z. L. Zhang, R. Tsang and D. Du, Adaptive Proportional Routing: a Localized QoS Routing Approach, IEEE/ACM Transactions on Networking, 10 (2002), pp. 790--804.

[8] S. Alabbad, and M. E. Woodward, “Localized Credit Based QoS Routing", Proceedings of IEE, vol. 153, PP. 787-796, 2006

[9] S. Alabbad, and M. E. Woodward, “Localized Credit Based QoS Routing: Performance Evaluation Using Simulation”, Proceedings of the 40th Annual Simulation Symposium, IEEE Huntsville, AL April 2-6, 2006.

[10] A. Medina, A. Lakhina, I. Matta and J. Byers, Brite: An approach to universal topology generation, Proceedings of the International Workshop on Modeling Analysis and Simulation of Computer and Telecommunications Systems, Cincinnati, Ohio, 2001, pp. 346-356.