A Survey of QoS Routing Algorithms

Marilia Curado, Edmundo Monteiro

Abstract— There is an obvious need for Quality of Service (QoS) on the Internet and QoS routing is an important component of the overall QoS framework. The role of a QoS routing strategy is to compute paths that are suitable for the different types of traffic generated by the various applications, while maximizing the utilization of network resources. The fulfilment of these objectives requires the development of algorithms that find multi-constrained paths taking into consideration the state of the network and the traffic requirements, namely, considering its needs in terms of delay, jitter, loss rate and available bandwidth. However, the problem of finding multi-constrained paths has high computational complexity, and thus there is the need to use algorithms that address this difficulty. This paper presents and discusses the main approaches used to reduce QoS routing algorithm complexity and to improve the overall network performance.

Keywords—Complexity, QoS routing algorithms.

I. Introduction

THE path computation algorithm is at the core of QoS routing strategies. Instead of using a shortest path algorithm based on statically configured metrics, as in traditional routing protocols, the algorithm must select several alternative paths that are able to satisfy a set of constraints regarding, for instance, end-to-end delay bounds and bandwidth requirements. However, the algorithms to solve such a problem have been shown to have, in general, high computational complexity. Several approaches have been proposed to address the complexity of multi-constrained path computation problem. The selection of QoS paths subject to multiple constraints can be defined as the Multi-Constrained Path (MCP) problem. In order to present the definition of the MCP problem, some definitions are introduced, as follows: a network is represented by a directed graph *G(V,E)* composed of a set of vertices (*V*) and a set of edges (*E*). The number of vertices of *G* is given by $n = |V|$ and the number of edges is given by $m = |E|$. Each edge, is represented by the link between two vertices $e = (u,v)$ and has associated *q* weights corresponding to QoS metrics such that $w_i(u,v) >= 0$, and $i = 1,2,\ldots,q$. The constraint for each QoS metric is L_i. The *Multi-Constrained Path problem* is to find a path *P* from a source *s* to a destination *d* such that all the QoS constraints are met, as depicted in the following equation:

$$w_i(P) \le L_i, \qquad i = 1, 2, \ldots, q \tag{1}$$

The paths that satisfy these constraints are called *feasible paths* [1]. The solution of the MCP problem requires a path computation algorithm that finds paths that satisfy all the constraints as expressed in Equation 1. Since the optimal solution of this type of problems for multiple additive and independent metrics is NP-complete, usually heuristics or approximation algorithms are used.

The first approach considered is used when bandwidth is one of the constraints that must be satisfied by the path computation algorithm. In this case, the MCP problem is defined as a Bandwidth Restricted Path (BRP) problem [2].

The second approach is called Restricted Shortest Path (RSP) and is a simplification of the original MCP problem, when two additive metrics are used [1]. In this case, all the paths that satisfy the constraint associated with one of the metrics are computed and then the best path according to the second metric is selected.

Metrics Combination (MC) is the third approach for the solution of the MCP problem [3]. By combining a set of QoS metrics in a single metric, it is possible to use existing path computation algorithms, such as Bellmand-Ford or Dijkstra.

Algorithms that solve the multi-constrained path problem using the above strategies are described in the following sections.

II. Bandwidth Restricted Paths

Bandwidth is widely used as a metric for QoS routing, alone or associated with other metrics, such as delay or number of hops [4]. The utilization of bandwidth in association with other metric simplifies the original MCP due to the fact that it is a concave metric that has a non-cumulative composition rule over a path. *Metric Ordering* is one of the main heuristics used for the solution of the BRP problem. This heuristic requires the identification of the metric that has higher priority and the computation of the best paths according to this metric. Afterwards, it is computed the best path according to the second metric. The algorithms that solve the BRP problem using metric ordering are the Widest-Shortest Path (WSP) and Shortest-Widest Path (SWP) algorithms. In these families of algorithms, the width of a path is depicted by the available bandwidth and its length can correspond either to the number of hops or to delay. The nature of the delay that is used for

Manuscript received November 30, 2004. This work was supported in part by the Portuguese Ministry of Science and High Education (MCES), and by European Union FEDER under program POSI (Project QoS II and IPQoS) and under the PhD grant PRAXIS XXI/ BD/13723/97.

Marilia Curado and Edmundo Monteiro are with the Laboratory of Communications and Telematics, CISUC/DEI, Pólo II, Pinhal de Marrocos 3030-290 Coimbra, Portugal (phone: +3512397900; fax: +351239701266; e-mails: marilia@ dei.uc.pt, edmundo@dei.uc.pt).

path computation, namely, propagation, transmission, queuing delay, depends on the specific algorithm.

A. Widest-Shortest Path algorithms

The objective of Widest-Shortest Path algorithms is to select the shortest path that is a feasible path according to the bandwidth constraint of flows. The main metric considered in WSP algorithms is the number of hops, and the second metric is available bandwidth. The algorithm proceeds as follows. In the first stage all existing shortest paths between each source and all destinations in the network are computed. In the second stage, bandwidth is used to break ties among paths that have the same number of hops, and it is selected the path that has the highest amount of available bandwidth. Widest-Shortest Paths can be computed by modified versions of Bellman-Ford or Dijkstra algorithms. Extensions to the Bellman-Ford algorithm, named Iterative Bellman-Ford, were presented by Ma and Steenkiste [5] and Apostolopoulos *et al.* [6]. Ma and Steenkiste proposed a modification of the Dijkstra algorithm for the computation of Widest-Shortest Paths [5]. Extensions to the Dijkstra algorithm for the computation of Widest-Shortest Paths were also done by Sobrinho [7].

The main performance objective of WSP algorithms is to reduce *network cost*. Network resource consumption is limited because the shorter paths are favored over wider paths. Since resource preservation is especially important when the network is congested, this type of algorithm shows very good performance when the network load is high. Widest-Shortest Path algorithms also show good behavior when the routing decision is taken upon information of the state of the network that is inaccurate [5, 8]. This is due to the fact that the main metric used (number of hops) is not significantly influenced by the inaccuracy of routing information since it changes less often than available bandwidth. WSP performs well because it is simultaneously conserves resources by choosing the shortest path and does load balancing by choosing the widest path among those that have the same length.

B. Shortest-Widest Path algorithms

The objective of Shortest-Widest Path algorithms is to find the paths with the highest amount of bandwidth available, the widest path. In a second phase, if there are several paths with the same amount of available bandwidth, it is selected the shortest path according to the length metric used, either number of hops or end-to-end delay. Ma and Steenkiste proposed extensions to the Dijkstra and Bellman-Ford algorithms to compute SWP paths where the path length is measured by the number of hops [5]. Wang and Crowcroft also proposed extensions to the Dijkstra and Bellman-Ford algorithms to compute SWP paths. However, in their approach called Delay Aided Bandwidth Search (DABS), the path length corresponds to the propagation delay of the path [5]. The DABS algorithms perform path pre-computation in a distributed manner with complexity similar to the complexity of the originating algorithm. However, the DABS algorithm based on the Dijkstra algorithm has been shown incapable of computing Shortest-Widest Paths in hop-by-hop routing [5, 7]. Despite the previous results, the DABS algorithm can be used to successful compute Shortest-Widest Paths in source routing.

A source routing variant of DABS has been used in the first phase of the MAximally DiSjoint WIdest Paths (MADSWIP) [9]. This algorithm uses a combined approach of path pre-computation and on-demand path computation in order to select maximally disjoint paths that satisfy flow QoS requirements in a connection oriented network.

Load balancing is the main performance objective of Shortest-Widest Path algorithms. This is achieved through the selection of the path with the highest availability of bandwidth. The main advantages of SWP algorithms are simplicity and pre-computation[1] capability. The former allows for scalability and the latter allows for a reduced flow setup time. SWP algorithms have, however, two main disadvantages. The first pertains to the increase in network cost because, in general, the widest path corresponds to a longer path [2, 5]. The second pertains to the degradation of traffic performance, due to the selfish behavior of the path computation algorithm. Therefore, the utilization of the widest path can restrict the admission of flows with more strict requirements, which could otherwise be avoided by computing a path with just the enough available bandwidth to satisfy the requirements of the flow. This approach would however require a more complex algorithm.

C. Hybrid algorithm

A compromise between the two performance objectives of WSP and SWP algorithms is achieved by the All Hops Optimal Path (AHOP) algorithm [10]. The AHOP algorithm is a variant of the shortest-widest algorithm family that tries to reduce network cost while achieving load balancing. The AHOP algorithm uses an extended version of the Iterative Bellman-Ford algorithm to pre-compute the minimum hop count path that meets the bandwidth constraints of the flow. The algorithm computes alternative shortest paths between the source and all destinations with increasing hop count. The shortest path is chosen if it can satisfy the request of the flow in terms of available bandwidth. Otherwise, it is chosen the shortest possible path that has enough available bandwidth. This means that a longer path is only used if it has more available bandwidth. Since the proposed solution uses hop count as the main metric, it is able to minimize network cost, while providing load balancing among admitted flows.

III. Restricted Shortest Path

The Restricted Shortest Path problem is a special case of the MCP problem when two additive metrics are used. This class of routing problems is solved by algorithms that find feasible paths according to one of the constraints and from those it computes an optimal path according to the other constraint, if such a path exists. A widely studied case of the RSP problem

[1] While path pre-computation reduces flow setup time, it may lead to the use of stale routing information and produce paths that are not able to support the requested QoS.

group is the Delay-Constrained Least Cost problem (DCLC). The definition of the DCLC problem is the following [11]: Each edge has two associated non-negative metrics: a cost, *c(e)* and a delay, *d(e)*. Given a source node *s* and a destination node *d*, *P(s, d)* is the set of all paths from *s* to *d*. The cost and delay of path P_i is given by the following equation:

$$c(P_i) = \sum_{e \in P_i} c(e) \quad and \quad d(P_i) = \sum_{e \in P_i} d(e) \tag{2}$$

P'(s, d) is the subset of *P(s, d)* that satisfies the delay constraint Δ_{delay} as depicted in the following equation:

$$d(P_i) \leq \Delta_{delay} \tag{3}$$

From this subset, the path that minimizes the cost is the delay-constrained least cost path according to:

$$\min_{P_i \in P'(s,d)} c(P_i) \tag{4}$$

The DCLC problem is NP-complete. The main algorithms used to solve this problem are described next.

The Delay Constrained Unicast Routing (DCUR) algorithm proposed by Salama and Reeves uses a heuristic to compute delay constrained least cost paths in a distributed manner [11]. The DCUR algorithm is based on a Distance-Vector algorithm and has a complexity of $\Theta(n^3)$.

Chen and Nahrstedt proposed a heuristic to solve the RSP problem called Delay-Cost Constrained Routing (DCCR) [12]. The heuristic reduces the complexity of the initial problem by modifying the cost function, making a transformation of the real valued metrics in integer values, therefore restraining the search space granularity. An example concerning the transformation of the delay metric is presented in Equation 5, where *d(e)* is the original delay of link *e*, Δ_{delay} is the constraint relative to delay, *coef* is a positive integer and *d(s,t)* is the number of hops between the source node *s* and the destination node, *d*. *nd(e)* is the new integer value concerning delay.

$$nd(e) = \left\lceil \frac{d(e).coef.d(s,t)}{\Delta_{delay}} \right\rceil \tag{5}$$

Afterwards, a source routing shortest path algorithm, either extended Dijkstra or extended Bellman-Ford, is used to compute the end-to-end delay and cost constrained paths.

Cheng and Ansari propose a source routing algorithm called Dual Extended Bellman-Ford (DEBF) to solve the DCLC problem [13]. The heuristic used by the DEBF algorithm is a bi-directional application of the Extended Bellman-Ford (EBF) algorithm previously proposed by the authors. In the forward phase the EBF is applied to compute the least cost feasible path from the source to the destination using the delay metric. If a path that satisfies the delay constraint is found on the forward phase, the cost metric is used in the backward phase. Since the DEBF algorithm executes the EBF twice, it has time complexity of $\Theta(2nm)$.

IV. Metrics combination

Metrics combination reduces the complexity of the RSP problem by combining both metrics in a single metric and then using a traditional shortest-path algorithm to compute the path that minimizes the resulting metric. Metrics composition: can use linear, non-linear, and Lagrange relaxation composition.

The combination of additive metrics using linear functions has been used in several proposals, namely the Multi-constrained Energy Function based Pre-computation Algorithm (MEFPA) [16] and simplified for two metrics in the Jaffe Approximation Algorithm (JAA) [15] and in the QoS Routing strategy of the University of Coimbra [17]. For additive metrics, each represented by w_i, for an edge *e*, the composition rule is expressed in Equation 8, where a_i is the relative weight of metric w_i. The link weights are computed through linear energy functions, where each energy function is a weighted sum of the link metrics, as depicted in Equation 8.

$$l(e) = \sum_{i=1}^{k} a_i w_i, \quad a_i \in [0,1], \quad \sum_{i=1}^{k} a_i = 1 \tag{6}$$

Even tough metrics combination contributes to the simplification of path computation algorithms it has, however, the drawback of preventing the provisioning of guarantees regarding the constraints associated with each one of the metrics involved. In order to overcome this problem, there is the need to define the proper weights used in the combination rule of metrics. This can be achieved by using *Lagrange relaxation* techniques to define the weights of the composition function [18]. Korkmaz *et al.* proposed the Binary Search for Lagrange Relaxation (BSLR) algorithm that uses a refined Lagrange relaxation technique to define the weights of the metrics composition rule [19]. In BSLR a binary search is performed to find the adequate value of *k*. It uses a hierarchical version of the Dijsktra algorithm, applied iteratively with evolving metrics, which computes shortest paths according to both metrics and keeps information concerning the total weights of the best shortest path of each metric.

Jüttner *et al.* proposed the LAgrange Relaxation based Aggregated Cost (LARAC) algorithm is a polynomial time solution for the delay constrained least cost routing problem [20]. The metrics considered are delay and number of hops, represented respectively by *d(P)* and *c(P)*. Since the number of hops is minimized, network resource utilization is optimized. The delay and cost metrics are combined in a single cost function shown in Equation 7:

$$c_\lambda(P) = c(P) + \lambda d(P) \tag{7}$$

Afterwards, the Lagrange multiplier is adjusted in consecutive iterations of the Dijkstra algorithm using the above defined cost. The running time complexity of the algorithm is $\Theta(m^3 log^4 m)$ and even though the algorithm does not give an exact solution, it always gives a bound for the solution. However, the linear combination of metrics may originate a metric whose minimization may not lead to a feasible solution, especially if the two metrics are not co-related [14, 21]. As an alternative, a non-linear function may be used, as in the next two algorithms.

The Delay-Cost Constrained Routing (DCCR) algorithm finds a near optimal solution to the Restricted Shortest Path problem. DCCR and its extension, the Search Space Reduction DCCR (SSR-DCCR) compute a path that satisfies delay and cost bounds, if such a path exists. In a first stage, a cost bound is defined according to network state. The weight of a path *P* from node *s* to node *u* is given by Equation 8, where *d(P)* and *c(P)* are the delay and cost of path *P*, respectively.

$$w(P) = \begin{cases} \dfrac{d(P)}{1 - c(P)/\Delta_{\cos t}}, \\ \quad if\ d(P) \le \Delta_{delay} \wedge c(p) \le \Delta_{\cos t} \\ \quad \infty,\ otherwise \end{cases} \tag{8}$$

Afterward, this non-linear combination of path delay and cost are used by the K-Shortest Path algorithm to search for a path that satisfies both constraints. This algorithm finds a list of *k* paths with the minimum cost, for a given source-destination pair in a graph. K-Shortest Path algorithms store multiple shortest paths at each node in increasing weight order.

Neve and Mieghem proposed the Tunable Accuracy Multiple Constraints Routing Algorithm (TAMCRA) [14]. Metrics are combined according to Equation 9 and a k-Shortest Path algorithm us used to compute the shortest path between the source and the destination.

$$l(P) = \max(c(P)/\Delta_{\cos t}, d(P)/\Delta_{delay}) \tag{9}$$

Lagrange relaxation based algorithms have low time complexity and results in the literature show that they can very often achieve either feasible or optimal solutions. Feng *et al.* make an evaluation of algorithms that use Lagrange relaxation to solve the Delay Constrained Least Cost problem using both linear and non-linear cost functions [22]. The approaches using combined metrics have strengths and weaknesses, specifically, the combination of metrics in a single metric allows for simple and well known path computation algorithms, however, the rule for combination of the metrics is not always straightforward.

V. CONCLUSIONS

The need of Quality of Service on the Internet has motivated the development of several mechanisms to evolve actual IP networks. Quality of Service routing is one of these components and was the main subject of this paper, namely in the subject of path computation algorithm complexity.

QoS routing has as main objective the selection of paths that satisfy the requirements of traffic in the network, while contributing to improved network resource utilization. The main problem to be solved by QoS routing algorithm is the Multi-Constraint Path problem. Algorithms to solve this family of problems are kwon to heuristics to reduce the complexity of the path computation problem, however, at the expense of not attaining the optimal solution for the problem, finding just a feasible solution. Within this framework, the QoS routing algorithms presented were grouped in the categories of Bandwidth Restricted Path algorithms, Restricted Shortest Path algorithms and algorithms that use metrics combination and the advantages and disadvantages of each approach were highlighted.

REFERENCES

[1] F. Kuipers, T. Korkmaz, M. Krunz, P. Van Mieghem, "Overview of Constraint-Based Path Selection Algorithms for QoS Routing", IEEE Communications Magazine, special issue on IP-Oriented Quality of Service, December 2002.
[2] Q. Ma, P. Steenkiste, "Routing Traffic with Quality-of-Service Guarantees in Integrated Services Networks", Proceedings of Workshop on Network and Operating Systems Support for Digital Audio and Video, July 1998.
[3] A. Jüttner, B. Szviatovszki, I. Mécs, Z. Rajkó, "Lagrange Relaxation Based Method for the QoS Routing Problem", IEEE INFOCOM 2001, Anchorage, Alaska, April 22-26, 2001.
[4] Z. Wang, J. Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications", IEEEJSAC, vol. 14. No 7, September 1996.
[5] Q. Ma, P. Steenkiste, "Quality-of-Service Routing for Traffic with Performance Guarantees" Proceedings of IFIP Fifth International Workshop on Quality of Service, Columbia University, New York, May 1997.
[6] G. Apostolopoulos, D. Williams, S. Kamat, R. Guérin, A. Orda, T. Przygienda, "QoS Routing Mechanisms and OSPF Extensions", IETF, RFC 2676, August 1999.
[7] J. L. Sobrinho, "Algebra and algorithms for QoS path computation and hop-by-hop routing in the Internet", IEEE Transactions on Networking, August 2002.
[8] Y. Jia, Y. Nikolaidis, P. Gburzynski. "Alternative Paths vs. Inaccurate link State Information in Realistic Network Topologies", Proceedings of SPECTS'02, pp. 162-169, San Diego, CA, July 14-18, 2002.
[9] N. Taft-Plotkin, B. Bellur, R. Ogier, "Quality-of-Service Routing Using Maximally Disjoint Pahs", Proceedings of IWQoS 1999, London, June 1-4, 1999.
[10] R. Guérin, A. Orda, "Computing Shortest Paths for Any Number of Hops", IEEE/ACM Transactions on Networking, October 2002.
[11] H. Salama, D. Reeves, Y. Viniotis, "A Distributed Algorithm for Delay-constrained Unicast Routing", INFOCOM'97, Japan, March 1997.
[12] S. Chen, K. Nahrstedt, "On Finding Multi-Constrained Paths", Proceedings of IEEE International Conference on Communications (ICC 98), pp. 874-879, Atlanta, GA, June, 1998.
[13] G. Cheng and N. Ansari, "A New Heuristics For Finding The Delay Constrained Least Cost Path", Proceedings of IEEE GLOBECOM 2003, San Francisco, CA, USA, pp. 3206 – 3210, Dec. 1-5, 2003.
[14] H. De Neve, P. Van Mieghem, "TAMCRA: A Tunable Accuracy Multiple Constraints Routing Algorithm", Computer Communications, Vol. 23, 2002.
[15] J. Jaffe, "Algorithms for finding paths with multiple constraints", Networks, vol. 14, pp. 95-116.
[16] Y. Cui, J. Wu, "Precomputation for Multi-constrained QoS Routing in High-speed Networks", Proceedings of IEEE INFOCOM 2003, March 30 – April 3, San Francisco, USA, 2003.
[17] M. Curado, J. Brito, B. Melo, G. Quadros, E. Monteiro, "Quality of Service Routing in the Differentiated Services Framework", Proceedings of SPIE's International Symposium on Voice, Video, and Data Communications, Boston, MA, USA, November 5-8, 2000.
[18] S. Boyd, L. Vandenberghe, "Convex Optimization", Cambridge University Press, 2004.
[19] T. Korkmaz, M. Krunz, S. Tragoudas, "An efficient algorithm for finding a path subject to two additive constraints", Joint International Conference on Measurement and Modeling of Computer Systems, Proceedings of the 2000 ACM SIGMETRICS, Santa Clara, California, United States, 2000.
[20] A. Jüttner, B. Szviatovszki, I. Mécs, Z. Rajkó, "Lagrange Relaxation Based Method for the QoS Routing Problem", IEEE INFOCOM 2001, Anchorage, Alaska, April 22-26, 2001.
[21] L. Guo, I. Matta, "Search Space Reduction in QoS Routing", 19th IEEE International Conference on Distributed Computing Systems, Austin, Texas, USA, May 31 - June 04, 1999.
[22] G. Feng, C. Douligeris, K. Makki, N. Pissinou, "Performance Evaluation of the Delay-Constrained Least-Cost Routing Algorithms Based on Linear and Nonlinear Lagrange Relaxation", Proceedings of Fiftieth IEEE International Conference on Communication (ICC 2002), pp 2273-2278, April 2002.