International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 02 | Feb 2021 www.irjet.net p-ISSN: 2395-0072

Exact Algorithms for Multi–Constrained Quality of Service Routing

Dr.Ibrahim Mohamed Ahmed Ali[1], Prof.G.K.Viju[2]

[1]Assistant Professor, Karary University, Khartoum, Sudan
[2]Professor and Dean (E-Learning), University of Garden City, Khartoum, Sudan

---

Abstract - *QoS routing is expected to be an essential building block of a future, efficient and scalable QoS-aware network architecture. Multi–constrained Quality of Service routing finds a route in the network that satisfies multiple independent qualities of service constraints. This paper concentrates on algorithms to solve the multiconstrained routing (MCP) problem. Exact QoS algorithms, on the other hand, guarantee that a path satisfying user needs would be found and offer a more realistic approach for solving the MCP problem. In this paper SAMCRA (**Self Adaptive Multiple Constraints Routing Algorithm**) has been used as an exact QoS routing algorithm that guarantees to always find the shortest path from source to destination. SAMCRA is exact this means that there is no other algorithm that can guarantee quality of service.*

Key Words: *Look-ahead, path dominance, QoS routing, shortest path, SAMCRA, MCP, Quality of Service.*

1. INTRODUCTION

Network routing essentially consists of two identities, the routing protocol and the routing algorithm. The routing protocol supplies each node in the network with a consistent view of that topology and, in some cases, of its resources at some moment in time. In the Internet is decomposed into two levels: (1) intra-domain routing, where routing is performed within a network, and (2) inter-domain routing, for routing between networks. The current dominant intra-domain routing protocol is called Open Shortest Path First (OSPF). Based on this information, the Dijkstra algorithm is used at each node to compute the shortest to all other nodes in the network. Quality of Service is the ability of a network element to have some level of assurance that its traffic and service requirements can be satisfied."

In unicast quality-of-service (QoS) routing, we wish to find a path from a source vertex to a destination vertex of a network. This path must satisfy specified constraints. The problem of determining a QoS route that satisfies two or more path constraints (for example, delay and cost) is known to be NP-hard. Many papers have targeted the QoS routing problem, but only a few dealt with the general MCP problem. Jaffe [1] proposed a shortest path algorithm using a linear combination of the link weights. Iwata *et al.* [2] proposed a polynomial-time algorithm to solve the MCP problem. The algorithm first computes one (or more) shortest path(s) based on one QoS measure and then checks if all the constraints are met. If this is not the case, the procedure is repeated with another measure until a feasible path is found or all QoS measures are examined. Chen and Nahrstedt [3] provided two approximate algorithms for the MCP problem. The algorithms return a path that minimizes the first (real) weight provided that the other weights are within the constraints. Korkmaz and Krunz [4] have proposed a randomized heuristic for the MCP problem. Under the same network conditions, multiple executions of the randomized algorithm may return different paths between the same source and destination pair. Hence the focus has been on the development of heuristics[7] and exact algorithms for the construction of multiconstrained QoS paths.

Multiconstrained Routing: Consider a graph in which each link u->v from node u to node v is characterized by a m dimensional link weight vector where the component is a QoS measure such as delay, jitter, loss, minimum bandwidth, cost, etc. The QoS routing algorithm computes the path that obeys multiple constraints, for all.

Definition 1*:Multi-Constrained Path (MCP) problem:* Consider a network G = (V, E).Each link (u→v€E) is specified by a link weight vector with as components m additive link weights wi (u→v)> = 0 for all 1<= i< = m. Given m constraints Li, where 1 = i <= m, the problem is to find a path P from a source node s to a destination node t such that

$$\overrightarrow{W}(p) = \sum_{j=1}^{h} \overrightarrow{W}(n_j, n_{j+1}) \quad \ldots (1)$$

for all 1<=i<=m.

A path that satisfies all m constraints is referred to as a feasible path. There may be many different paths in the graph G that satisfy the constraints. According to definition 1, any of these paths is a solution to the MCP problem. The problem that additionally optimizes some length function l(.) is called the multi-constrained optimal path problem.

***Definition** 2: Multi-Constrained Optimal Path (MCOP) problem:*[5] Consider a network G = (V, E). Each link (u→v) € E is specified by a link weight vector with as components m additive QoS link weights WI (u→v) >= 0 for all 1< = i< = m. Given m constraints Li, where 1 <= i <= m, the problem is to find a path P from a source node s to a destination node t satisfying (1) and, in addition, minimizing some length criterion such that l (P) < = l (P 0), for all paths. Korkmaz and Krunz [5] also provided a heuristic called H MCOP. This heuristic tries to find a path within the constraints by using the nonlinear path length function of SAMCRA [6]. Both heuristics of Korkmaz and Krunz apply some form of the look-ahead function .Yuan [7] presents two heuristics for the MCP problem, which are similar to TAMCRA [8], but use a Bellman–Ford approach. Liu and Ramakrishnan [9] considered the problem of finding not only one but multiple shortest paths satisfying the constraints. Many heuristics have been proposed for this problem, e.g., see [11], [12], [13], and [14]. In addition, new algorithms were proposed to find more than one feasible path w.r.t. bandwidth and delay. Specifically, if weighted fair queuing scheduling is being used and the constraints are on bandwidth, queuing delay, jitter, and loss, then the problem can be reduced to a standard shortest path problem by representing all the constraints in terms of bandwidth.

2. MATERIALS AND METHODOLOGY

PATH LENGTH L (P): The *weight* of a path vector is a vector sum. The definition of the path length is needed to be able to compare paths since the link weight components all reflect different QoS measures with specific units. Linear path length as proposed by jaffe [1]

$$l(P)= \sum_{i=1}^{m} d_i w_i (p) = d \cdot \vec{w}(p) \quad \text{..... (2)}$$

where di are positive real numbers. When scanning the solution space with a straight equilength line l(P) = l as in Fig.1, the area scanned outside the constraint region is minimized if the slope of the straight equilength lines satisfies (d1)/(d2)=(L2)/(L1).

The nonlinear definition is

$$L_q (P) = \left(\sum_{i=1}^{m} [\underline{w_i}(P)]^q\right)^{1/q}$$

The Dijkstra algorithm applied to the reduced graph will return a “shortest” path. In spite of the advantage that the simple Dijkstra shortest path algorithm can be used, the drawback is that the shortest path does not necessarily satisfy all constraints.

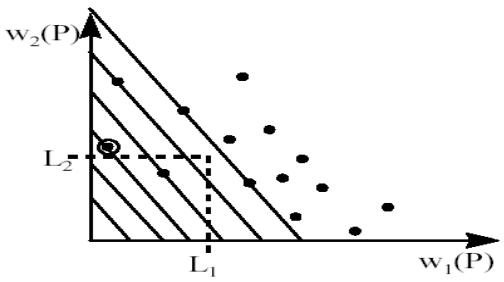

Fig.1.Illustration of straight equilength line

In the example graph of Fig. 2 for the constraints (14, 11, 22), the shortest path from node a to node i runs over node c, e and f. according to the definition (3), the path length of P (i - f - e - c - a) equals 0.95, which means that it satisfies all constraints. However, as illustrated in Fig.2 the shortest path from node a to node e does not traverse node c but node b.

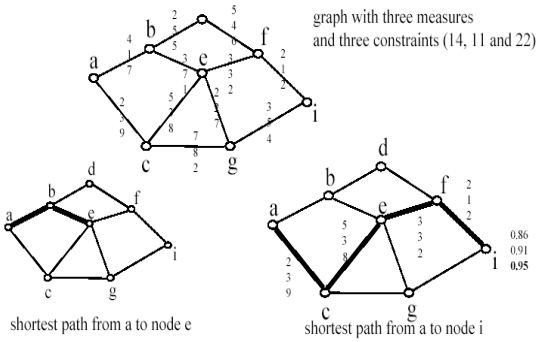

Fig.2. Illustration of the property that, when using a nonlinear path length definition, subsections of shortest paths or not necessarily shortest paths.

K-SHORTEST PATH ALGORITHM: A k-shortest path algorithm[9] is similar to Dijkstra's algorithm. Instead of storing at each intermediate node only the previous hop and the length of the shortest path from the source to that intermediate node, it can store the shortest, the second shortest, the third shortest... up to the k-shortest path together with the corresponding length. It is possible to store less than k paths at a node, but not more. The value of k can be limited as in SAMCRA'S companion TAMCRA [8].There exist many k-shortest path

DOMINATED PATHS: If the value of k in a k-shortest p based algorithm is unrestricted, then it is necessary to reduce the search space[11] to increase the computational efficiency.

In fig.3 (a), P1 is shorter than P2 and wi (P1) < wi (P2) for all 1 = i = m components. In that case, any path from the source to the final destination node that uses P1 will be shorter than any other path from this source to that destination that makes use of P2. Indeed, if, for all i, wi (P1) = wi (P2), then wi (P1) + ui = wi (P2) + ui for any ui.The value of (P1) only stored in the queue.

In fig.3 (b), both paths have crossing abscissa and ordinate points: wi (P1) <wi (P2) for some indices i, but wj (P1) > wj (P2) for at least one index j. Hence, if two subpaths have crossing abscissa-ordinate values, all m components of both paths must be stored in the queue.

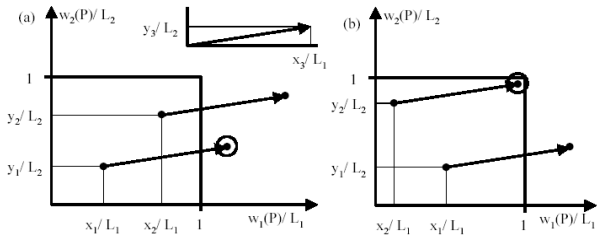

Fig.3.Dominated Paths (a), P1 dominates P_2, but in (b), neither P_1 nor P_2 is dominant.

LOOK-AHEAD: Besides path dominance, the look-ahead concept can be viewed as an additional mechanism to reduce the search space[11] of possible paths. The look-ahead concept proposes to compute the shortest path tree rooted at the destination to each node n in the graph G for each of the m link weights separately. Hence, for each link weight component 1< = i <= m, the lowest value from the destination to a node n is stored in the queue of that node n. The basic importance of look-ahead is to provide each node n with an exact, attainable lower bound of, for each individual link weight component i.

In Fig. 4, SAMCRAv1 searches the shortest path starting from the neighbors of the source node the two new lengths of the subpaths will be stored in the queue of the nodes 1 and 2, respectively. The next subpath to be extracted is the one with lowest length. Thus, the subpath in the queue of node 1 is extracted and its neighbors are scanned. The new subpath with length 0.8 is stored in node 3 .The next shortest subpath with length 0.3 is extracted and after scanning the neighbors, a new subpath is stored in node 3 with length 0.4. This is also the shortest subpath and it is consequently extracted from the queue of node 3. Finally, the destination node is reached and SAMCRAv1 stops.

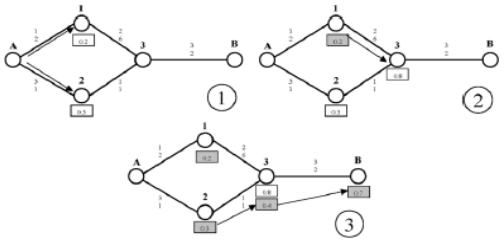

Fig. 4 Original operation of samcra without look-ahead improvement.

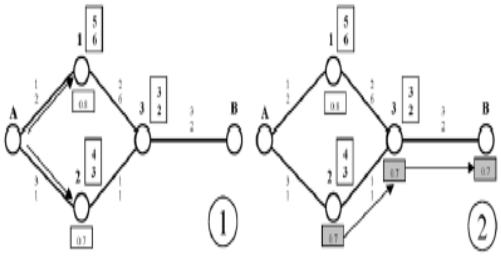

Fig.5 .SAMCRA with look-ahead.

In Fig. 5, first the Dijkstra algorithm for each measure has been executed from destination node to all the nodes. The lower bounds obtained are drawn in a rectangular box at the right of the node number. Similarly, as in the SAMCRAv1 operation, the search starts from the source node to its neighbors. Two new sub paths are found, but this time the "predicted" length is stored instead of the real length. Hence, for node 1, a new sub path with length 0.8 is stored and for node 2 one with 0.7. The next sub path to be extracted is from node 2 and colored in grey. It has one neighbor in node 3. The predicted length 0.7 is stored in the queue of node 3. This is still the shortest length and, hence, also extracted from node 3. Finally, the destination is reached from node 3 with a shortest length of 0.7.

SAMCRA ALGORITHM: The previous sections have listed four concepts for an exact and efficient QoS routing algorithm. All four concepts are present in SAMCRA.

Meta-code SAMCRA

INITIALIZE (G, m, A, B, L)

While (Q≠Ø)

EXTRACT-MIN (Q→u[i])

u[i] ←GREY

If (u=B)

STOP return path

else

for each v€Adj[u]\{$\prod$[u[i]],A)

FEASIBIITY(G,u,I,v,counter,d,w,maxlength)→dominated

Predicted length←l(d[u[i]]+w(u→v)+b[v])

If (predicted_length≤maxlength AND dominated≠ 1)

UPDATEQUEUE(Q,u,I,v,j,d,w, ╥,counter[v],predicted length)

If (v=b AND predicted_length<maxlength)

maxlength←predicted_length

3. HOW IT WORKS

The network is characterized by two parameters: delay and costs. The example shows a search for the path between node A and node B that complies with the requirements, in this case set at 10,10. That means the delay must be less than 10 units and the costs must not exceed € 10. In fig.6 the initialisation stage the Dijkstra's algorithm is used to find the lower limit for the distance from each node to the destination. These lower limits bounds are shown in the rectangles. In fig.7 SAMCRA then starts to choose the path with the shortest length and extends this path to the neighbors under certain conditions. SAMCRA's length function is non-linear and it is shown in the small rectangles. Once a path has been chosen it is colored grey. Step 3 shows that various paths can be established for a node. In fig.8 SAMCRA stops (step 6) when the path with the Shortest path is reached.

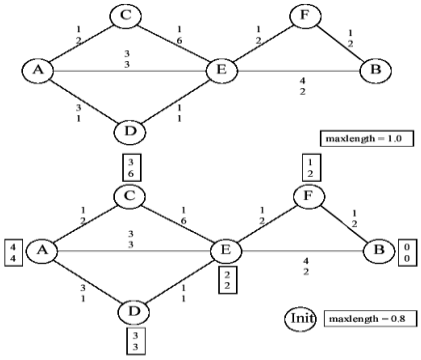

Fig .6 Example of operation of SAMCRA

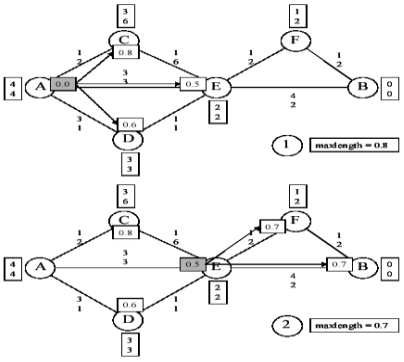

Fig .7 Example of the operation of SAMCRA (step 1 and 2)

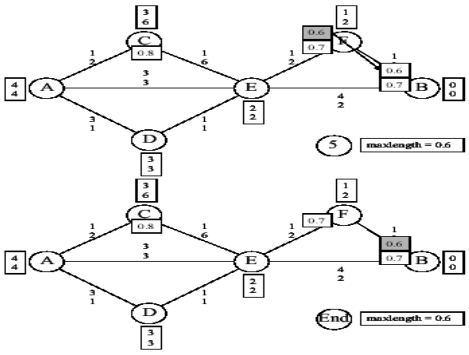

Fig.8 Example of the operation of SAMCRA (Step 5 and step 6) .

4. COMPARISON WITH EXISTING ALGORITHMS

In this paper the simulation with Waxman graphs and lattices is used . The weights of a link were assigned independent uniformly distributed random variables in the range [0, 1]. Then also simulated with two negatively correlated QoS measures, for which the link weights were assigned. During all simulations, the success rate and the normalized execution time were stored. The success rate of an algorithm is defined as the number of times that an algorithm returned a feasible path divided by the total number of iterations. The normalized execution time of an algorithm is defined as the execution time of the algorithm divided by the execution time of Dijkstra's algorithm. In Fig 9 gives the success rate and normalized execution time for the class of Waxman graphs and lattices, with m = 2. The exact algorithms SAMCRA and A*Prune always give success rate = 1. In Fig 10 also displays the normalized execution time that the algorithms needed to obtain the corresponding success rate. For the class of lattices the execution times of the exact algorithms grow exponentially.

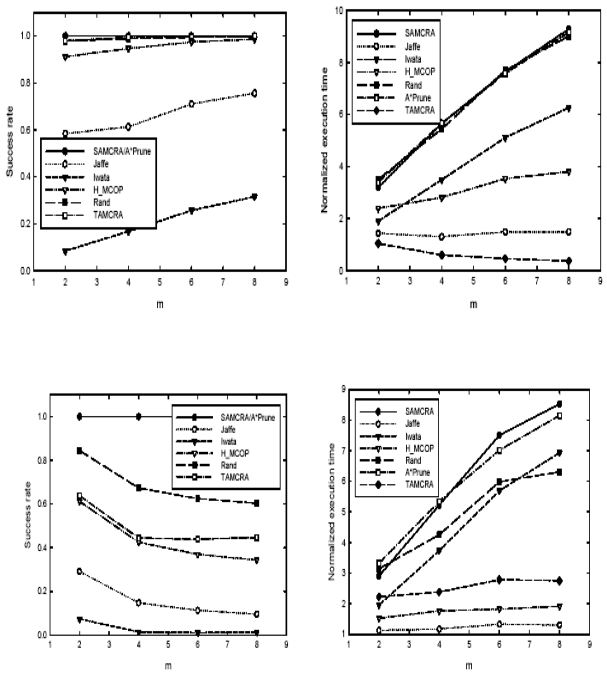

Fig.9,10 The success rate and normalized execution time for the class of Waxman graphs (above) and lattices (below) as a function of the number of nodes.

5. CONCLUSION

Four concepts for exact QoS routing were identified: a non-linear length function, the k-shortest paths approach, the principle of non-dominance and the look-ahead technique. The multiple QoS constraints make any length function non-linear. A problem of the inherent non-linearity is that subsections of shortest paths are not necessarily shortest themselves, which necessitates to evaluate multiple paths at a node. This is accomplished by a k-shortest paths[9] approach. In the worst case this k-shortest paths approach could evaluate all possible paths and therefore efficient search-space-reducing[11] techniques are required. Two such techniques are "the principle of non-dominance" and "the concept of look-ahead. "Look-ahead computes lower-bound vectors to assist in determining whether a path can become feasible or not. Based on these four concepts here proposed the exact algorithm SAMCRA. Based on such new concepts SAMCRA could evolve over time and maintain the top-position it has acquired today.

REFERENCES

1. J. M. Jaffe, "Algorithms for finding paths with multiple constraints, "Networks, vol. 14, pp. 95–116, 2010.
2. A. Iwata, R. Izmailov, D.-S. Lee, B. Sengupta, G. Ramamurthy, and H.Suzuki, "ATM routing algorithms with multiple QoS requirements for multimedia internetworking," in IEICE Trans. Commun. E79-B, vol. 8, 2006, pp. 999–1006.
3. S. Chen and K. Nahrstedt, "On finding multi-constrained paths," in Proc. IC Conf., New York, NY, 2008, pp. 874–879.
4. T. Korkmaz and M. Kuntz, "A randomized algorithm for finding a path subject to multiple QoS requirements," Comput. Networks, vol. 36, pp. 251–268, 2011.
5. "Multi-constrained optimal path selection," in Proc. IEEE INFOCOM, vol. 2, 2001, pp. 834–843.
6. P. Van Mieghem, H. De Neve, and F. Kuipers, "Hop-by-hop quality of service routing," Comput. Networks, vol. 37, no. 3–4, pp. 407–423, 2001.
7. X. Yuan, "Heuristic algorithms for multiconstrained quality-of-service routing," EEE/ACM Trans. Networking, vol. 10, pp. 244–256, Apr.2002
8. H. De Neve and P. Van Mieghem, "TAMCRA: A tunable accuracy multiple constraints routing algorithm," Comput. Commun, vol. 23, pp.667–679, 2000.
9. G. Liu and K. G. Ramakrishnan, "A*Prune: An algorithm for finding K shortest paths subject to multiple constraints," in Proc. IEEE INFOCOM, vol. 2, 2001, pp. 743–749.
10. Z. Wang and J. Crowcroft, "Quality-of-service routing for supporting multimedia applications," IEEE J. Select. Areas Commun. vol. 14, no. 7, pp. 1228–1234, Sept. 1996.
11. L. Guo and I. Matta, "Search space reduction in QoS routing," in Proc.19th Int. Conf. Distributed Computing Systems, III, May 1999, pp. 142–149.
12. R. Hassling, "Approximation schemes for the restricted shortest path problem," Math. Oper. Res., vol. 17, no. 1, pp. 36–42, 1992.
13. S. Reeves and H. F. Salama, "A distributed algorithm for delay constrained unicast routing," IEEE/ACM Trans. Networking, vol. 8, pp.239–250, Apr. 2000.
14. A. Orda, "Routing with end-to-end QoS guarantees in broadband networks,"IEEE/ACM Trans. Networking, vol. 7, pp. 365–374, June 1999.