COMPARING A NOVEL QOS ROUTING ALGORITHM TO STANDARD PRUNING TECHNIQUES USED IN QOS ROUTING ALGORITHMS

Wayne S. Goodridge[1] wgoodrid@dal.ca
William Robertson bill.robertson@dal.ca
Bill Phillips bill.phillips@dal.ca
Shyamala Sivakumar sivas@dal.ca

[1] Dalhousie University, Dpt. of Enginerring Mathematics, [2] St. Mary's University

Abstract

The problem of finding QoS paths involving several combinations of network metrics is NP-complete. This motivates the use of heuristic approaches for finding feasible QoS paths. Many constraint based routing algorithms find QoS paths by first pruning resources that do not satisfy the requirements of the traffic flow and then running a shortest path algorithm on the residual graph. This approach results in a QoS path that biases the first metric used in the search process. In addition, it can be shown that this approach may not always find the optimal path. Our research introduces a QoS routing algorithm that is based on a decision support system that is used to compute QoS paths. We demonstrate the feasibility of this approach by comparing it to standard pruning techniques.

Keywords: *Decision Support System, QoS routing, pruning techniques*

1. INTRODUCTION

Traditional link state and distance vector routing protocols such as OSPF always forward packets to the shortest path. This can cause problems for flows with a need for QoS guarantees if the shortest path does not have enough resources to meet the requirements. Both IntServ and Diffsev provide mechanisms for flows to reserve resources on the shortest path. However, the shortest path cannot make the reservation if there are not sufficient resources along the path to begin with. What is missing is a framework that can find a path, if one exists, which has the requested resources available. QoS routing is a routing scheme that considers the quality of service requirements of a flow when making routing decisions.

Dynamic determination of feasible paths is one of the key objectives of QoS Routing. That is, given a set of QoS requirements of a flow the goal is to find a path that can accommodate these requirements. A QoS routing algorithm should achieve this objective by optimization of resource usage. This is a very important objective since if the QoS routing process chooses a path that is more desirable than the user-specified QoS then the user does not gain any additional utility and network resources are wasted. On the other hand, if a path is not found that meets the user-specified QoS then this may affect the user application performance. Hence, a routing paradigm that emphasizes searching for an acceptable path satisfying various QoS requirements is needed in integrated communication networks.

Wang and Crowcroft in [1] stated that "The problem of finding a path subject to constraints on two or more additive and multiplicative metrics in any possible combination is NP-complete." Some efforts [2, 3] have been made to simplify the situation and can be categorized as: (1) Single Metric expressed as a linear combination of weighted link metrics. The resultant path is optimal in terms of the single metric and not optimal in terms of the individual metrics - that is, information is lost in the aggregation process; and (2) Pruning the network where link constraints that do not meet the requested resources links are deleted from the topology. This will guarantee that any path found on the pruned topology satisfies the link constraint in question. The Widest-Shortest Path and Shortest-Widest Path (SWP) algorithms use pruning. The resulting path usually is a function of the first metric since the second metric is used only when the first metric could not determine the best path.

In this paper we introduce a novel technique called Routing Decision Support System (RDSS) that computes QoS paths for given flow requirements. In Section 2 we describe the RDSS algorithm and in Section 3 we describe how the algorithm can be used to solve constraint and optimization problems. In Section 4 we present a simulation to compare the SWP and RDSS algorithms.

CCECE 2004- CCGEI 2004, Niagara Falls, May/mai 2004
0-7803-8253-6/04/$17.00 ©2004 IEEE

2. RDSS Algorithm

To tackle the NP- complete problem of finding QoS paths based on multiple metrics we decided to formulate a solution to this problem as follows: (1) finding all the fixed paths connecting edge devices in an un-weighted graph with the same links; and, (2) feeding the found paths from step 1, along with the flow demand $\{f_{m1}, f_{m2}, \ldots f_{mm}\}$ into a decision support system (DSS) which uses a generic multiple metric algorithm to obtain the QoS path in polynomial time. To achieve this we employed the concept of a preference function [4] that is concerned with ordering both preferences and differences of criteria used in the decision making process. To build a preference function for a given criterion involves gathering the values for that metric on each route and then constructing a scale. In our solution each scale has a range set between -1 and 1 and a scale is constructed using the following algorithm:

1. For a given metric say m_j select all metric values m_{ij} for routes R_i, where $i = \{1,,2,...,N\}$
2. For a given metric m_j the best value from the set of all routes for m_j is assigned to a_{ij} where i is the route corresponding to the best value for metric m_j.
3. For a given metric m_j the worse value from the set of all routes for m_j is assigned to b_{ij} where i is the route corresponding to the worse value for metric m_j.
4. A preference scale is then derived for each criterion with the following properties:
   a. $s : A \rightarrow F$ where $F \subseteq \Re$ (not defined) and $F \subseteq [-1,1]$
   b. A is bounded such that and $s(b) = -1$ and $s(a) = 1$
   c. For concave metrics if $x < b$ then $s(x)=-1$ and if $x > a$ and $s(x)=1$.
   Additionally, for additive and multiplicative metrics if $x < b$ then $s(x)=0$ and if $x > a$ and $s(x)=1$.

   d. All other values of x are mapped by s as follows:

$$s(x) = \left(2\frac{(x-b)}{(a-b)} - 1\right) \quad (1)$$

5. Let A be a n x m matrix (m = number of criteria (route metrics), n = number of alternative routing paths) containing columns $s_1(x_1)$, $s_2(x_2)$, ..., $s_m(x_m)$ where $s_j(x_j)$ represents a scale for each metric m_j. The preference function for routes R_i is therefore given by $A\overline{v} = \overline{o}$ where o is an N x 1 vector and v is a M x 1.

When a router receives a flow demand request the following procedure is followed:

1. Each value in the flow demand is converted to its corresponding scale value as follows:
   $d = [d_1 \quad d_2 \quad \ldots \quad d_m]$,
   $f(d) = [f(d_1), f(d_2) \ldots f(d_3)]$

   and
   $v = [s_1(f(d_1)) \quad s_2(f(d_2)) \quad \ldots \quad s_n(f(d_n))]$
   where f is a function that returns the closest largest definable point on the scale s. For example, if s is defined for points (2,4,8,20) and the argument of f is 6.5, the largest definable point in the domain of s is 8, and therefore s(8) will be evaluated rather than s(6.5).

2. The $A\overline{v} = \overline{o}$ matrix multiplication is then performed which results in an N x 1 vector. Vector $\overline{o}$ is examined for the highest value of the vector at position y where $1 \le y \le m$. The value of y corresponds to the route that will be considered by the negotiation phase for QoS admission of flow demand d.

3. For the negotiation phase each value of the flow demand is checked against the path metrics for route y. If any value d in the flow demand is not met and the s(d) = -1, then d ● a, and steps 1 and 2 are repeated. If all path metrics for route y are satisfied then the flow is accepted, otherwise the flow is rejected.

2.1 RDSS Example

To illustrate how the RDSS algorithm works consider the directed network in Figure 1. Suppose that network traffic flows use ingress router A and egress router F. Table 1 shows all the possible paths connecting routers A and F and the corresponding delays and jitters for each path. Now suppose a user request the demand d = [delay = 7, jitter = 4].

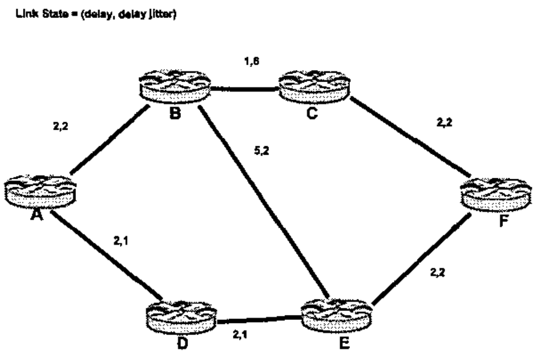

Fig. 1. Example Network

Table 1. Paths values for network metrics

Path	Delay	Jitter
P(A,B,C,F)	5	10
P(A,D,E,F)	6	4
P(A,B,E,F)	9	6
P(A,D,E,B,C,F)	12	12
a	5	4
b	12	12

$$\begin{bmatrix} 1 & -0.5 \\ 0.71 & 1 \\ -0.14 & 0.5 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 0.43 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.07 \\ 1.31 \\ 0.43 \\ -1.43 \end{bmatrix}$$

Since 1.31 is the largest value the route corresponding to this position is the solution, namely A,D,E,F. Examining Figure 1 confirms that this is the best path to satisfy the given constraints.

3. RDSS as a Solution

Contributions in [5] outlined definitions of link-constrained link-optimization routing, multi-link-constrained routing, link-constrained path-constrained routing, link-optimization path-constrained routing, multi-path-constrained routing and path-constrained path-optimization routing problems. Relative straightforward solutions to all these problems are outlined in [5] with the exception of the multi-path-constrained routing and path-constrained path-optimization routing problems that are NP-complete. All solutions involve appropriate variations of pruning links and then running modified versions of Dijkstra's algorithm.

The RDSS algorithm solves all multi metric constraint problems by simply setting the values of d_j for each metric m_j. However, constraint-optimization problems are more difficult to solve using this technique and this will be demonstrated in the following section.

4. Simulation

To evaluate our solution to finding suitable QoS paths for given flow demands we implemented both the RDSS algorithm and the SWP algorithms in Java. We then ran both algorithms with 10000 bandwidth values randomly generated between 0 and 40. Both algorithms then optimize the delay constraint. In the case of the RDSS this is done by setting $s(d_1)$ to a small non-negative value. Figure 2 shows the network topology used in the simulation.

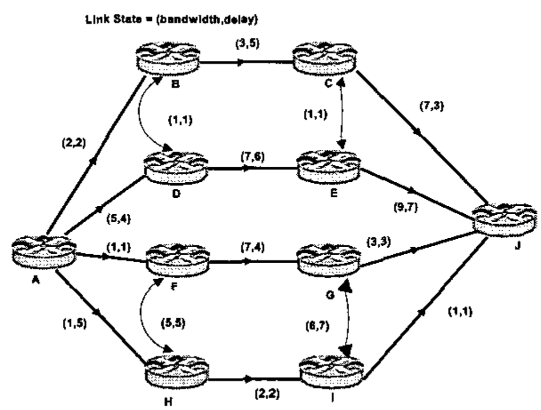

Fig. 2. Network Topology

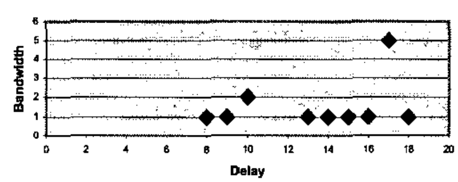

Fig. 3. Path Constraints for the Simulated Network

Figure 3 shows a scatter plot for all the path-constraints. Figure 4 shows the selected paths when the RDSS algorithm is used while Figure 5 shows selected paths when SWP is used. Figure 6 and 7 show the successful constraints for runs with the RDSS SWP algorithms.

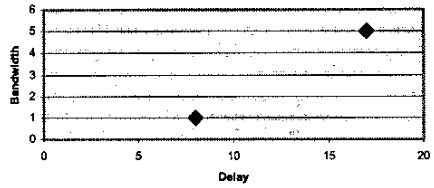

Fig. 4. Accepted Paths for RDSS algorithm

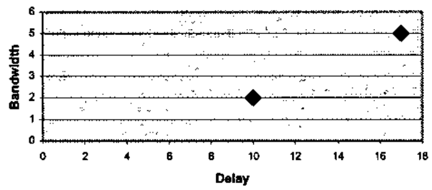

Fig. 5. Accepted Paths for SWP algorithm

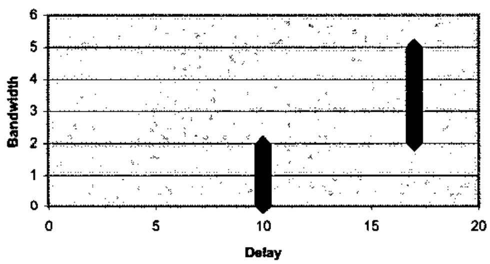

Fig. 6. Successful Constraints for SWP algorithm

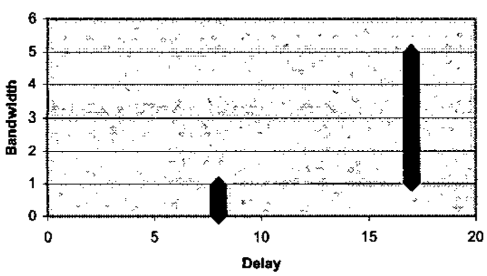

Fig. 7. Successful Constraints for RDSS algorithm

6. CONCLUSIONS

The results suggest that the SWP does not always find the most optimal path for the delay metric even when such a path exists as illustrated in Figures 4, 5, 6 and 7. On the other hand the RDSS found an optimal path for the delay metric for bandwidth values between 0 and 1 Mbps and between 2 and 40 Mbps. However, it is clear that the RDSS algorithm has drawbacks when solving constraint-optimization problems since it did not find an optimal path for bandwidth values between 1 Mbps and 2 Mbps when the algorithm selected a path with bandwidth constraint of 5 Mbps and delay of 17 ms rather than one with bandwidth constraint of 2 Mbps and delay of 10 ms.

The RDSS algorithm offers a generic one-stop solution for multiple metric constraint-based problems. However, we have to find a more precise way to handle multiple metric constraint-optimization problems.

References

[1] Z. Wang and J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications", IEEE Journal on Selected Areas in Communications, vol. 14, n. 7, pp. 1228-1234, 1996.

[2] G. Apostolopoulos, et al, "Intra-Domain QoS Routing in IP Networks: A feasibility and Cost/Benefit Analysis". IEEE Network, 13(5):42-54, September-October 1999

[3] S. Chen and K. Nahrstedt, "An Overview of Quality–of–Service Routing for the Next Generation High-Speed Networks: Problems and Solutions," *IEEE Networks*, Special Issue on Transmission and Distribution of Digital Video, Nov./Dec., 1998.

[4] J. Barzilai, "Notes on measurement and Decision Theory," Proceedings of the NSF Design and Manufacturing Research Conference, San Juan, Puerto Rico, pp.1-11, 2002

[5] Ilmari Juva, "Analysis of Quality of Service Routing Approaches and Algorithms ", HELSINKI UNIVERSITY OF TECHNOLOGY, Thesis 2003.