A MULTICONSTRAINT QOS ROUTING SCHEME USING A MODIFIED DIJKSTRA'S ALGORITHM

DONG-WON SHIN
CERIAS and School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907-1285, USA
E-mail: shin3@purdue.edu

EDWIN K. P. CHONG[†] AND HOWARD JAY SIEGEL[§]
Department of Electrical and Computer Engineering[†§]
Department of Mathematics[†]
Department of Computer Science[§]
Colorado State University, Fort Collins, CO 80523-1373, USA
E-mail: {echong,hj}@colostate.edu

We propose a multiconstraint QoS routing scheme, MPMP (multi-prepaths multi-postpaths). MPMP achieves low EDR and polynomial worst-case time complexity using a modified Dijkstra's algorithm with a metric, called the minimum normalized margin. We show by simulation that MPMP has better performance than two competing schemes from the literature, TAMCRA and H_MCOP.

1 Introduction

Multiconstraint QoS (quality of service) routing is an essential mechanism to support computer/communication applications that have requirements on various QoS attributes. The multiconstraint QoS routing problem involves finding a *feasible* path, i.e., a path satisfying a given set of QoS constraints between given source and destination nodes. Unfortunately, the multiconstraint QoS routing problem is known to be NP-complete [1]. Hence, several heuristic schemes have been proposed for its solution. Typical performance measures for such schemes are time complexity and erroneous decision rate (EDR). *EDR* is defined as the fraction of instances that a routing scheme either fails to find a feasible path that exists, or finds a path that turns out to be infeasible. Low time complexity and low EDR are the goal of multiconstraint QoS routing schemes.

In this paper, we propose a multiconstraint QoS routing scheme, called MPMP (multi-prepaths multi-postpaths). MPMP searches for a feasible path

This research was supported in part by the Center for Education and Research in Information Assurance and Security (CERIAS), by the Colorado State University George T. Abell Endowment, and by NSF under grants 0098089-ECS and 0099137-ANI.

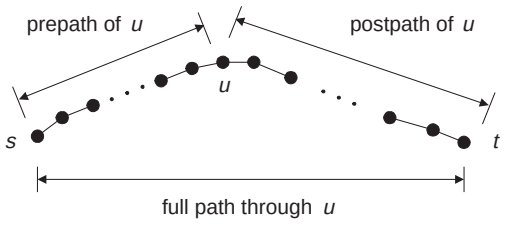

Figure 1. A prepath and a postpath of node u, and a full path through node u when s and t are source and destination (terminal) nodes, respectively.

using a modified Dijkstra's algorithm with a new metric, called the minimum normalized margin (NM_{min}). The NM_{min} measures the severity of constraints, and reduces the run time of MPMP by detecting the paths that cannot be extended to a feasible path, eliminating them from consideration.

Henceforth, we call any path between a given pair of source and destination nodes a *full path*. In addition, for an arbitrary node u, we call any path from the source node to u a *prepath* of u, and any path from u to the destination node a *postpath* of u, as illustrated in Figure 1. *MPMP* is a routing scheme that selects multiple prepaths and multiple postpaths for every node. In MPMP, any single-constraint shortest-path algorithm (e.g., Dijkstra's or the Bellman-Ford algorithm) can be used to select postpaths at the first step of the routing procedure. Based on the selected postpaths, MPMP uses a modified Dijkstra's algorithm to select prepaths for each node during the routing procedure. MPMP has polynomial worst-case time complexity (assuming a fixed maximum number of prepaths per node), and can trade off the chance of finding a feasible path with the memory space for storing prepaths. We show by simulation that MPMP has better performance than two competing schemes from the literature, TAMCRA [2] and H_MCOP [3].

2 Multiconstraint QoS Routing

2.1 Multiconstraint QoS Routing Problem

We assume that the following are given: a network topology; a set of values associated with each link with respect to additive QoS attributes [1]; and a connection request indicating a source node, a destination (terminal) node, and the constraints that the routing path must satisfy. We also assume that there is at most one link between any two nodes, that the network topol-

ogy does not change throughout the routing procedure, and that every QoS attribute value is nonnegative and fixed.

Because we assume that there is at most one link between any two nodes, we can represent any link by its two endpoint nodes. We denote the link between (arbitrary) nodes u and v by (u, v). To represent a path between two arbitrary nodes, we list all the nodes on the path between '$\langle$' and '$\rangle$'. With this notation and all of the above assumptions, we can formulate the multiconstraint QoS routing problem as follows.

Definition 1 (Multiconstraint QoS Routing Problem) *Suppose we are given an undirected graph representing a network topology, $G = (V, E)$, where V and E are the sets of n nodes and m links, respectively. Suppose also that the link (u, v) between arbitrary nodes u and v is characterized by nonnegative values of q additive QoS attributes, $d_j(u, v) \geq 0, j = 1, \ldots, q$. Given a source node s, a destination node t, and a constraint value C_j with respect to each QoS attribute for $j = 1, \ldots, q$, the multiconstraint QoS routing problem is to find a path $p = \langle s, w_p(1), \ldots, w_p(b-1), t\rangle$, where $w_p(i), i = 1, \ldots, b-1$, is a node on p, such that the length of p with respect to the jth QoS attribute, i.e., $L_j(p) = d_j\big(s, w_p(1)\big) + d_j\big(w_p(1), w_p(2)\big) + \ldots + d_j\big(w_p(b-1), t\big)$, is less than or equal to the corresponding constraint value C_j for every $j = 1, \ldots, q$.*

2.2 Nonlinear Path Length

One approach to solving the multiconstraint QoS routing problem is to use an extended version of a single-constraint shortest-path algorithm. However, the "length" of a path in the multiconstraint QoS routing problem cannot be defined as in the single-constraint shortest-path problem. We can resolve this difficulty by introducing the notion of nonlinear path length. Let C_j and $L_j(p)$ for $j = 1, \ldots, q$ be as defined in Definition 1. Recall $L_j(p)$ is the length of path p with respect to the jth QoS attribute. The *nonlinear path length*[α] of p, denoted by $\Lambda(p)$, is defined as follows [2]:

$$\Lambda(p) = \max\Big(\frac{L_1(p)}{C_1}, \frac{L_2(p)}{C_2}, \ldots, \frac{L_q(p)}{C_q}\Big). \tag{1}$$

It is straightforward to see that p is feasible if and only if $\Lambda(p) \leq 1$. Therefore, the nonlinear path length provides a basis for using single-constraint shortest-path algorithms for the multiconstraint QoS routing problem. However, standard shortest-path algorithms rely on the property that the length

[α]The nonlinear path length is referred to as just *path length* in [2]. However, we use the qualifier "nonlinear" to distinguish it from the path length with respect to a single QoS attribute.

of a path is the sum of quantities associated only with individual links on the path, a property that fails to hold for the nonlinear path length. Hence, some modification to the standard approach is necessary, as described next.

An arbitrary prepath p of an intermediate node u is *nondominated* [4] if there is no other prepath of u that has a smaller QoS attribute value for every QoS attribute than p. If p is dominated, then p cannot be the subpath of the shortest full path. A brute-force algorithm using the notion of nonlinear path length for multiconstraint QoS routing maintains a set for each node that stores all nondominated prepaths found so far [5,6]. When the algorithm terminates, we get all nondominated full paths, and thus we can find the shortest full path and check its feasibility. This brute-force algorithm has exponential complexity, which requires an impractically long run time and an excessive amount of memory for large networks.

Neve and Mieghem propose a modification to the brute-force algorithm, called TAMCRA [2], by limiting the number of prepaths per node. During the course of the routing procedure, TAMCRA stores for each node at most k shortest (in terms of nonlinear path length) prepaths that have been found so far, hoping that they would have higher probability than other prepaths to be a subpath of the shortest full path through the node. When TAMCRA selects the prepaths for each node, it considers the nonlinear path lengths of the prepaths only. TAMCRA does not consider the length of the postpath that would be connected to the prepaths. Hence, if an arbitrary node u finds a new prepath p_n that has nonlinear path length smaller than its kth shortest prepath p_k, then p_n replaces p_k, even though p_n may be connected with a longer postpath than the one with which p_k would have been connected.

To overcome the drawback of TAMCRA mentioned above, Korkmaz and Krunz propose an enhanced scheme, called H_MCOP [3]. Different from TAMCRA, H_MCOP precomputes a single postpath for each node at the first step of the routing procedure, with the hope that this single postpath would be the subpath of a feasible path through the node. Then, H_MCOP uses the postpath to update the set of k prepaths with the goal that combining these prepaths with the postpath results in near-minimum nonlinear path lengths. However, if the selected postpath is not the subpath of an optimal path, then the postpath may misguide the selection of prepaths.

3 Metrics for MPMP

Assume that we wish to construct a path from source node s toward destination (i.e., terminal) node t, and that we have found a path p that reaches an arbitrary node u. Note that p is a prepath of u. Let b_p be the number of links

on p. Recall C_j, $L_j(p)$, $w_p(i)$, and $d_j(u,v)$ as defined in Definition 1. Then, the *normalized slackness* of p with respect to the jth QoS attribute, denoted by $NS_j(p)$, is defined as:

$$NS_j(p) = 1 - \frac{L_j(p)}{C_j} = 1 - \frac{1}{C_j} \sum_{i=1}^{b_p} d_j \left(w_p(i-1), w_p(i) \right), \tag{2}$$

where $w_p(0) = s$ and $w_p(b_p) = u$. Intuitively, $NS_j(p)$ is the fraction of C_j that is available to be "used" by a postpath connected to p. Note that $NS_j(p) \leq 1$.

Lemma 1 *Let p be a prepath of node u, and p' a postpath of u. Then, $p + p'$ is feasible if and only if $L_j(p')/C_j \leq NS_j(p)$ for every $j = 1, \ldots, q$.*

Proof: $p + p'$ is feasible if and only if $L_j(p) + L_j(p') \leq C_j$ for every $j = 1, \ldots, q$. Thus, $p+p'$ is feasible if and only if $L_j(p')/C_j \leq \left(1 - L_j(p)/C_j\right) = NS_j(p)$ for every $j = 1, \ldots, q$. ■

Let $H(u)$ be the set of all possible postpaths of node u, h an arbitrary path in $H(u)$, $w_h(i)$ the ith node along h, and b_h the number of links on h. The *normalized requirement* of u for the jth QoS attribute, denoted by $NR_j(u)$, is given by:

$$NR_j(u) = \min_{h \in H(u)} \frac{L_j(h)}{C_j} = \min_{h \in H(u)} \frac{1}{C_j} \sum_{i=1}^{b_h} d_j \left(w_h(i-1), w_h(i) \right), \tag{3}$$

where $w_h(0) = u$ and $w_h(b_h) = t$. Thus, $NR_j(u)$ is the minimum path length with respect to the jth QoS attribute (as a fraction of C_j) of any postpath of u. Note that $NR_j(u) \geq 0$.

Lemma 2 *Let p be a prepath of node u. If $NR_j(u) > NS_j(p)$ for some j, then no extension of p is feasible.*

Proof: Suppose $NR_j(u) > NS_j(p)$. For any postpath p' of u, because $L_j(p')/C_j \geq NR_j(u)$, $L_j(p')/C_j > NS_j(p)$. Thus, the result follows from Lemma 1. ■

The *normalized margin* of path p with respect to the jth QoS attribute, denoted by $NM_j(p)$, is the difference between $NS_j(p)$ and $NR_j(u)$, where p is a prepath of u (i.e., u is the endpoint node of p):

$$NM_j(p) = NS_j(p) - NR_j(u). \tag{4}$$

Hence, $NM_j(p)$ is the difference between the fraction of C_j remaining available (i.e., $NS_j(p)$) and the minimum fraction of C_j required for any postpath connected to p (i.e., $NR_j(u)$). Note that $NM_j(p) \leq 1$.

The *minimum normalized margin* of path p, denoted by $NM_{min}(p)$, is the smallest NM amongst all the NMs of p with respect to each QoS attribute,

where p is a prepath of u:

$$NM_{min}(p) = \min_{j=1,\ldots,q} NM_j(p) = \min_{j=1,\ldots,q} \Big[NS_j(p) - NR_j(u)\Big]. \qquad (5)$$

The quantity $NM_{min}(p)$ provides a measure of the severity of the strictest QoS constraint in searching for a feasible path including path p. A lower value of $NM_{min}(p)$ indicates that it is less likely to find a feasible path that extends path p. Note that $NM_{min}(p) \leq 1$.

Theorem 1 *If $NM_{min}(p) < 0$, then no extension of path p is feasible.*

Proof: This follows from eq. (5) and Lemma 2. ■

We first defined the concept of NM_{min} in our paper [7].

4 MPMP: Multi-Prepaths Multi-Postpaths

4.1 Basic Approach

MPMP builds upon the ideas of TAMCRA and H_MCOP to produce a scheme with smaller EDR without sacrificing run time. Like these schemes, MPMP uses an extended version of Dijkstra's algorithm. However, in contrast to them, MPMP selects multiple postpaths as well as multiple prepaths for each node, and uses NM_{min} as the metric for selecting prepaths. In MPMP, each selected postpath corresponds to the shortest postpath with respect to a given QoS attribute. Hence, that the number of postpaths selected per node is the same as the number of QoS attributes, and by eq. (3) the length of each postpath is the NR of the node with respect to the corresponding QoS attribute.

The MPMP algorithm is described in detail in the next section. At the heart of the MPMP algorithm is the procedure for updating the set of prepaths, based on the NM_{min} values. To explain this procedure, suppose we are given a set of prepaths. From this set, pick a prepath with the largest NM_{min}, and let u be its endpoint node (i.e., the node at the opposite end on the prepath to a given source node). Suppose that node v is an outgoing node (i.e., neighboring downstream node) of u, and that at most k prepaths have already been stored for u and v, respectively. By appending link (u, v), the prepaths of u can be extended into prepaths of v, resulting in up to $2k$ prepaths of v. MPMP has already found q postpaths for v at the beginning of the routing procedure. With these postpaths, MPMP can compute the NM_{min} for each of the above prepaths of v by eq. (5). Observe that MPMP does not need to store the q postpaths found at the beginning of the routing procedure; it only needs to store q NR values, one for each of these postpaths. (Note that H_MCOP also stores q values; however, these q values are

derived from a single postpath, in contrast to MPMP, where q postpaths are involved.) If the number of prepaths that MPMP can store for each node is limited to k, then MPMP selects the k prepaths with largest nonnegative NM_{min} values among the above prepaths. Prepaths with negative NM_{min} values are discarded, because they cannot be extended into feasible paths (by Theorem 1). This procedure is repeated for every outgoing node of u.

The selection of the prepath of a node with the largest NM_{min} maximizes the difference between what we have (i.e., NS) and what we must use (i.e., NR) for the most stringent constraint. Hence, MPMP uses a greedy approach in the sense that it leaves as much of the NM_{min} as possible for the remaining process of the routing procedure. The multi-postpath approach to compute NM_{min}s reduces the chance of selecting inappropriate prepaths leading to high EDR. In addition, discarding prepaths with negative NM_{min} reduces the run time of MPMP by eliminating them from consideration.

4.2 MPMP Algorithm

Pseudocode for the MPMP is shown in Figure 2. Let the maximum number of prepaths per node be k. Line 01 in the pseudocode computes q NR values for every node by q runs of any single-constraint shortest-path algorithm. On lines 02-03, MPMP checks if any NR of the source node is larger than 1, because no paths can be feasible in this case. Lines 04–06 represent the initialization procedure of the modified Dijkstra's algorithm, described on lines 07–24.

The modified Dijkstra's algorithm in MPMP maintains a set Q of prepaths whose nodes have already been determined. Q initially contains only a zero-length path from source node to itself. On lines 08–09, MPMP extracts from Q the prepath with the largest NM_{min} (denoted by x). If x is a full path, the routing procedure terminates (lines 10–11). Otherwise, on lines 12–14 MPMP extends x to every outgoing node v of the endpoint node of x. If the extended path (denoted by y) has a nonnegative NM_{min}, and if the number of prepaths of v in Q is less than k, then y is stored for v in Q (lines 15–18). However, if Q already has k prepaths for v, then MPMP compares the NM_{min} of y with the smallest NM_{min} among the NM_{min}s of the prepaths of v in Q (the path with the smallest NM_{min} is denoted by z). If y has a larger NM_{min}, it replaces z in Q (lines 19–23). If there is no prepath in Q, then the routing procedure terminates (line 24) with no feasible path found.

4.3 Complexity of MPMP

Let n, m, and q be the numbers of nodes, links, and QoS attributes, respectively. Line 01 in Figure 2 requires q executions of a single-constraint shortest-

```
MPMP(G = (V, E), s, t, q, k, C_1, ..., C_q)
01 compute NR_1(w), ..., NR_q(w) for every w ∈ V      /* using C_1, ..., C_q */
02 if any of NR_1(s), ..., NR_q(s) > 1
03    then return failure      /* no feasible paths */
04 for each w ∈ V
05    do n(w) ← 0      /* n(w): number of prepaths of w */
06 Q ← {⟨s⟩}      /* Q: set of prepaths with nodes determined */
07 while Q is not empty
08    do x ← path with largest NM_min in Q
09       Q ← Q − {x}      /* the selected prepath extracted from Q */
10       if end(x) = t      /* end(x): endpoint node of x */
11          then return x      /* a feasible path found */
12       for each v ∈ {w|(end(x), w) ∈ E}      /* outgoing node of end(x) */
13          do if v is not on path x      /* check looping*/
14             then y ← x + ⟨end(x), v⟩      /* x extended to end(x) */
15                if NM_min(y) ≥ 0 and n(v) < k
16                   then n(v) ← n(v) + 1
17                        end(y) ← v
18                        Q ← Q ∪ {y}      /* add y into Q */
19                else if NM_min(y) ≥ 0 and n(v) = k
20                   then z ← prepath of v with smallest NM_min in Q
21                        if NM_min(z) < NM_min(y)
22                           then end(y) ← v
23                                Q ← Q ∪ {y} − {z}      /* replace z with y */
24 return failure      /* no feasible paths */
```

Figure 2. Pseudocode for MPMP.

path algorithm. If we use Dijkstra's algorithm, then the run time of line 01 is $O(nq \log n + mq)$ [8]. If k prepaths of an arbitrary node u have been extracted on line 09, then no more prepaths of u can be added to set Q on line 18. Hence, Q contains at most kn prepaths. If we implement Q using a heap, then selecting a prepath with the largest NM_{min} among kn prepaths (line 08) takes $O(\log(kn))$ [8]. Inserting a prepath and heapifying Q on lines 18 and 23 also takes $O(\log(kn))$. Thus, the run time of lines 08, 18, and 23 is $O(kn \log(kn))$ for the entire course of the MPMP algorithm. Because at most k prepaths of each node are selected on line 08, the for-loop of lines 12–23 examines each link (u, v) at most k times in the adjacency lists of u and v, respectively. Hence, for the while-loop on lines 07–23, the total number of iterations of this for-loop is $O(km)$. Each run of this for-loop takes $O(k + n + q)$ without considering lines 18 and 23, because the checkup of loop, the computation of

$NM_{min}(y)$, and the selection of path z take $O(n)$, $O(q)$, and $O(k)$, respectively. Thus, the run time of the for-loop on lines 12–23 for the entire course of MPMP algorithm is $O(km(k+n+q))$ without considering lines 18 and 23. Therefore, by adding all these contributions, we obtain a worst-case time complexity for MPMP of $O(nq\log n+mq)+O(kn\log(kn))+O(km(k+n+q))$ $= O(nq\log n+kn\log(kn)+km(k+n+q))$. Note that if the maximum number of prepaths per node (i.e., k) is fixed, this time complexity is polynomial.

MPMP needs $O(kn^2)$ memory space for at most kn paths in Q, because each path has at most n nodes. In addition, $O(knq)$ memory space is needed to store the NRs, NMs, and NM_{min} for every path in Q. Hence, the memory complexity of MPMP is $O(kn(n+q))$. As the value of k increases, the EDR of MPMP decreases. Thus, MPMP achieves low EDR at the expense of the increased run time and the memory space for an increased number of prepaths.

5 Performance Evaluation

5.1 Generation of Network Topologies and QoS Attribute Values

To evaluate the performance of MPMP for the QoS routing problem with two QoS attributes, we perform 10,000 simulation runs. Each of these runs is executed according to the following steps. First, we generate two kinds of random network topologies using the Waxman model [9] and the Inet Topology Generator [10], respectively. A network topology generated by the Waxman model has 400 nodes and 1106 links, and an Internet-like network topology by the Inet Topology Generator has 4,000 nodes and 7,741 links.

Next, we assign QoS attribute values to every link in the generated network topologies. Specifically, each link has QoS attribute values independent of other links, and the pair of QoS attribute values of a link has a given correlation coefficient. Most of the previous papers on multiconstraint QoS routing assumed for simulation that each link had uniformly distributed random QoS attribute values within a limited range. Instead of fixing a range, we assume QoS attribute values to have a normal distribution. For every QoS attribute, we arbitrarily assume a unit mean, and set the variance as 0.16 to keep the probability of generating negative values less than 1%. When we generate a negative value, we replace it by zero. For the constraint value of every QoS attribute, we use 12 and 5 for the network topologies generated by the Waxman model and the Inet Topology Generator, respectively. These values have been chosen such that the constraints are not too loose or too stringent. The fraction of the simulation runs with no feasible path is approximately 10%.

Finally, we apply MPMP, TAMCRA, and H_MCOP for routing in each of

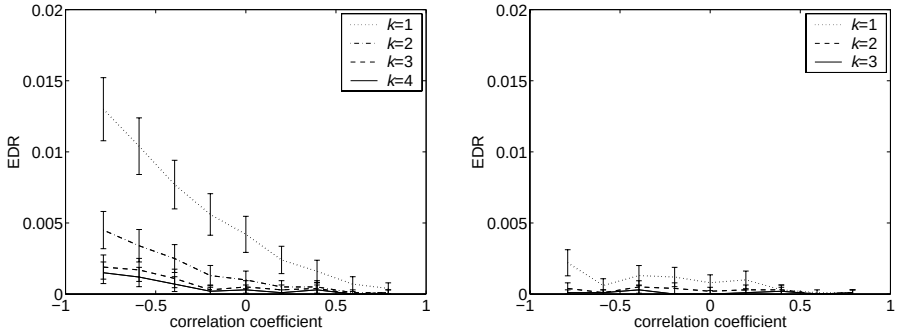

Figure 3. Plots of EDR versus correlation coefficient where MPMP is applied to (a) the 400-node network topologies generated by the Waxman model and (b) the Internet-like network topologies of 4,000 nodes. k denotes the maximum number of prepaths per node. No simulation for $k = 4$ in (b) was performed to reduce execution time. 95% confidence intervals are shown by the interval bars.

the network topologies. We also apply an exhaustive-search scheme to check if there exists any feasible path, for computing the EDR of each routing scheme.

5.2 Simulation Results

Figure 3 shows plots of EDR versus correlation coefficient for MPMP applied to the network topologies generated by the Waxman model and to the Internet-like network topologies. We can see that MPMP has low EDR for all the correlation coefficients, and that the EDR decreases as the maximum number of prepaths per node increases. Note that the EDR for the Internet-like network topologies is lower than for the network topologies generated by the Waxman model, despite the larger network size. The underlying reason is that the Internet-like network topologies have smaller average and larger variance of node degree than the network topologies generated by the Waxman model. Hence, the number of the prepaths for each node among which k prepaths are selected during the routing procedure is small, and thus MPMP rarely misses appropriate prepaths for each node.

Figure 4 shows plots of EDR versus correlation coefficient where MPMP, TAMCRA, and H_MCOP are applied to the network topologies generated by the Waxman model. We can see that MPMP has lower EDR than TAMCRA and H_MCOP. Note that TAMCRA has lower EDR than H_MCOP, in con-

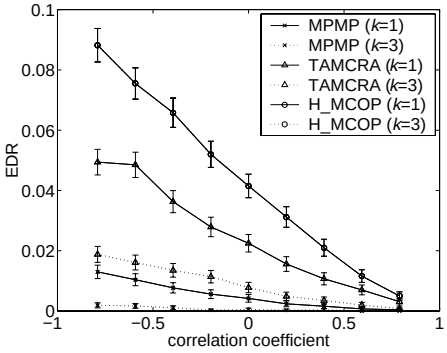

Figure 4. Plots of EDR versus correlation coefficient where MPMP, TAMCRA, and H_MCOP are applied to the 400-node network topologies generated by the Waxman model. k denotes the maximum number of prepaths per node. 95% confidence intervals are shown by the interval bars. Note that the plot lines for H_MCOP ($k = 1$) and H_MCOP ($k = 3$) are completely overlapped.

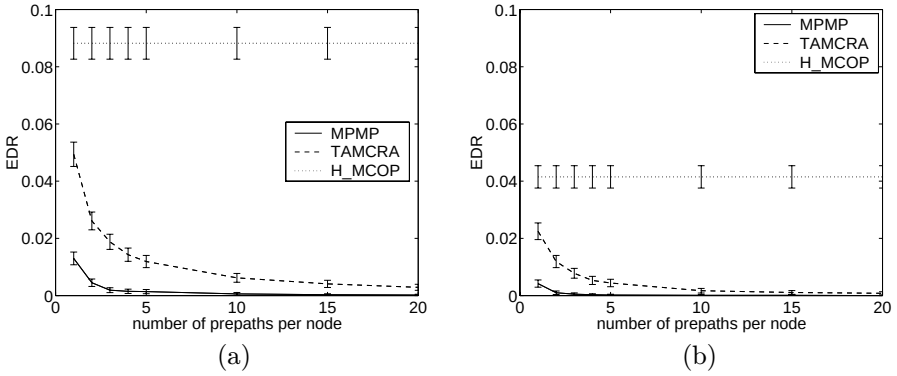

Figure 5. Plots of EDR versus maximum number of prepaths per node where MPMP, TAMCRA, and H_MCOP are applied to the 400-node network topologies generated by the Waxman model, when the correlation coefficients between two QoS attribute are (a) -0.8 and (b) 0, respectively. 95% confidence intervals are shown by the interval bars.

trast to the simulation result in [3]. We believe that this contrast is due to differences in the simulated network environments. Figure 5 shows plots of EDR versus maximum number of prepaths per node when the correlation coefficients are -0.8 and 0, respectively. We can see clearly from the plots that

MPMP has lower EDR than TAMCRA and H_MCOP for a fixed maximum number of prepaths per node. Thus, MPMP can achieve low EDR with a smaller amount of memory to store prepaths than TAMCRA and H_MCOP.

6 Conclusions

We have proposed a multiconstraint QoS routing scheme, MPMP. Like TAMCRA and H_MCOP, MPMP uses the notion of nonlinear path length. However, in contrast to them, MPMP selects multiple postpaths for each node, and uses NM_{min} as the metric to select prepaths for each node. These make it possible for MPMP to achieve low EDR and polynomial worst-case time complexity (assuming a fixed maximum number of prepaths per node). We showed by simulation that MPMP has better performance than TAMCRA and H_MCOP. MPMP provides a promising solution for multiconstraint QoS routing, which will become an essential tool in providing high-quality services for communication/computer systems.

Acknowledgments

The authors thank J. Kim and S. Ali for their helpful comments.

References

1. Z. Wang, *Information Processing Letters*, **69** (3), 111 (1999).
2. H. D. Neve and P. V. Mieghem, *Computer Communications*, **23** (7), 667 (2000).
3. T. Korkmaz and M. Krunz, *IEEE INFOCOM*, **2**, 834 (2001).
4. M. I. Henig, *European Journal of Operational Research*, **25** (2), 281 (1985).
5. J. Walrand and P. Varaiya, *High-Performance Communication Networks*, 2nd ed. (2000).
6. X. Yuan and X. Liu, *IEEE INFOCOM*, **2**, 844 (2001).
7. D. Shin, E. K. P. Chong, and H. J. Siegel, *IEEE Workshop on High Performance Switching and Routing*, 385 (2001).
8. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, 2nd ed. (1990).
9. B. M. Waxman, *IEEE Journal on Selected Areas in Communications*, **6** (9), 1617 (1988).
10. C. Jin, Q. Chen, and S. Jamin, CSE-TR-433-00, Department of Electrical Engineering and Computer Science, University of Michigan (2000).