An Effective Approximation Scheme for Multiconstrained Quality-of-Service Routing

Jun HUANG
Institute of Network Technology, Beijing Univ. of Posts & Telecom.
Beijing, P.R.China 100876
Email: xiaoniuadmin@gmail.com

Xiaohong HUANG
Institute of Network Technology, BUPT
Beijing, P.R.China 100876
Email: huangxh@buptnet.edu.cn

Yan MA
Institute of Network Technology, Beijing Key Lab of Intell. Telecom. Softw. & Multim., BUPT
Beijing, P.R.China 100876
Email: mayan@bupt.edu.cn

Abstract—Finding a path that satisfies multiple Quality-of-Service (QoS) requirements is vital to the deployment of current emerged services. However, existing QoS routing algorithms are not very efficient and effective at finding such path. Moreover, few works focus on two or more QoS constraints. In this paper, we propose an effective fully polynomial approximation scheme (FPAS) for multiconstrained path optimal problem based on the technique of auxiliary graph construction. By employing the nonlinear definition of the path constraint and limited iteration of the FPAS itself, our FPAS can not only achieve the complexity reduction but generate a preferable path as well. We further analyze the Markov properties of the entire network and obtain some key parameters to reflect the routing characteristic. We experiment with different scale of random networks and compare our FPAS against previous well known studies. Our results show that FPAS can find path with lower complexity and better quality.

I. INTRODUCTION

The last few years have witnessed the growth of various streaming media over the Internet, thanks to widespread deployment of the high-speed network technology. In parallel, real-time huge-volume data transfer applications (e.g. web broadcasting, video teleconferencing, and HDTV) also raise new challenges to current high-speed packet switching networks, one of the main issues is QoS routing, to select feasible paths to guarantee various applications to meet multiple quality-of-service (QoS) requirements [1].

Practically, QoS routing consists of two functionalities: routing protocols that keep the network state information up to date and QoS routing algorithms that compute the constrained shortest paths based on the information provided by the routing protocols. This paper mainly concerns with the latter one. The algorithms for computing paths can be used in many scenarios, for example, constructing label-switching paths in MPLS, arranging service-delivering paths in IMS-enabled networks, establishing wavelength-switching paths in fiber-optics networks based on the QoS requirements in the service contracts, or applying together with RSVP [2].

On the other hand, QoS metrics fall into two categories. The metrics of the path are obtained by adding (multiplying, in the case of reliability) the metrics of edges along the path, metrics such as delay, delay jitter, and hop count are additive. QoS metrics like bandwidth are known as bottleneck parameters where the corresponding metric of the path is the smallest of the metrics of the edges along the path. Since problems involving bottleneck constraints are solvable (by omitting all links from the topology that does not satisfy one of the constraints), the additive parameters are focused only in this paper.

The problem of QoS routing is challenging because selecting paths that meet two or more additive QoS constraints is an NP-hard [3]. Such fundamental problem is usually known as multi-constrained path problem (MCP). There has been much work involved in designing heuristic solutions for this problem. Sahni [4] presented some general techniques for combinatorial approximation. Chen et al. [2], [5] proposed a polynomial time algorithm for Delay Constrained Least Cost (DCLC) problem based on Sahni’s study. Xue [6] and Juttner et al. [7] presented the Lagrange relaxation method to approximate the DCLC by linear combination. Korkmaz and krunz [8] used a nonlinear function and proposed a randomized heuristic, their simulation results are rather good compared to other algorithms. In [9], Yuan presented a limited granularity heuristic and a limited path heuristic. Kuipers et al. [10] summarized the representative techniques for the DCLC. As one of the most important techniques, ϵ-approximation algorithm, which can obtain an approximate, quantifiable solution to the optimal one, was widely studied. Warburton [11] was the first to derive a fully polynomial approximation scheme (FPAS) for DCLC on acyclic graphs. Hassan improved the Warburton’s result with time complexity $O(|E| |V| (\frac{|V|}{\epsilon}) \log (\frac{|V|}{\epsilon}))$ in [12], where $|E|$ is the number of edges, $|V|$ is the number of nodes and $\epsilon$ is the approximation parameter. In addition, Lorenz et al. [13] presented a faster FPAS with a time complexity of $O(|E| |V| (\log \log |V| + \frac{1}{\epsilon}))$. More recently, Xue et al. [14], [15] studied the different version of the Multiconstrained Path (MCP) Problem and presented an efficient approximation algorithm, the proposed algorithm can solve the optimization version of MCP with K constraints in $O(|E| (\frac{|V|}{\epsilon})^{K-1})$ time.

To the best of our knowledge, Xue’s algorithm is currently the fastest one for path establishment in multiconstrained QoS routing. Nevertheless, the space complexity of their algorithm is relatively high, and its obtained path might not be precise enough. In this paper, we take a view of obtaining a more pre-
An example solution with lower complexity for the Multiconstrained Optimal Path (MCOP) is to enhance the performance of the network. More specifically, we extend Xue’s scheme which constructs an auxiliary graph to approximate the multiconstrained QoS routing problem. Differing from Xue’s study, we employ the nonlinear definition of the path constraints to reduce the complexity, and introduce the limited iterations of the algorithm itself to refine the precision of obtained solution continuously. We believe that the presented scheme could find a preferable \((1 + \epsilon)\)-approximation solution with much lower complexity. The theoretical analysis and the simulation study further confirm this conclusion.

The rest of this paper is organized as follows. Section II formally defines the problems and notations. The related work has also been reviewed in this section. Section III describes the proposed approximation algorithms. The analysis of routing properties based on proposed algorithms is conducted in Section IV. Section V presents the simulation results and Section VI draws the conclusion of this paper.

II. PRELIMINARIES

A. Problem Formulation

Our notation and terminology are from [14] and [16]. A communication network with \(M\) QoS constraints can be represented by a connected graph \(G(V, E, \varphi_1, \cdots, \varphi_M)\), where \(V\) is the set of \(|V|\) vertices, \(E\) is the set of \(|E|\) edges. Each edge has \(M\) weights, and \(\varphi_m(e) \geq 0\) is the \(m\)th weight of edge \(e\), \(\forall e \in E, 1 \leq m \leq M\). Weights on edges can straightforwardly extent to weights on paths. Let \(p\) be a path in \(G\). Denote \(\varphi_m(p)\) as the sum of the \(m\)th weight on edges along \(p\). We have following definitions.

Definition 1: Multiconstrained Path (MCP) Problem. Consider an undirected graph \(G(V, E)\). Each edge associated with \(M\) positive real-valued edge weights \(\varphi_m(e)\), \(1 \leq m \leq M\). Let \(\Psi = (\Psi_1, \Psi_2, \cdots, \Psi_M)\) be the \(M\) constraints, and \(s, d \in V\) be the source and destination nodes. Find a path \(p\) from \(s\) to \(d\) such that \(\varphi_m(p) \leq \Psi_m\).

The path that satisfied \(\varphi_m(p) \leq \Psi_m\) is said to feasible path. We use \(\{p_f\}\) to denote all the feasible paths in \(G(V, E)\).

Definition 2: Multiconstrained Optimal Path (MCOP) Problem. The MCOP problem is to find an optimal path \(p_{opt}\) among feasible paths \(\{p_f\}\) in \(G\) from \(s\) to \(d\) and smallest value of \(\eta \in (0, 1)\) such that \(\varphi_m(p_{opt}) \leq \eta \cdot \Psi_m\).

An algorithm is a \((1 + \epsilon)\)-approximation algorithm (or simply, an approximation algorithm) for MCOP if the algorithm generates a source to destination optimal path \(p_{opt}\) such that \(\varphi_m(p_{opt}) \leq (1 + \epsilon) \cdot \eta \cdot \Psi_m\), the running time of the algorithm is bounded by a polynomial in the input size of the instance, and in \(1/\epsilon\) as well.

B. The Main Idea of Xue’s Algorithm

Xue et al. provided an important methodology in their works [14], [15], that is, the construction of auxiliary graph. It transforms from an undirected graph \(G\) to a directed graph \(G^{M, \pi}\). Each vertex \(v \in G\) is associated with \((1 + \pi)^{K-1}\) vertices (since each weight has been normalized \(\frac{\varphi_m}{\Psi_m}\)), we could regard \(\pi\) as the maximum integer of all constraints.) in \(G^{M, \pi}\), i.e., \((u, \zeta_2, \cdots, \zeta_M)\), where \(\zeta_2, \cdots, \zeta_M \in [0, \pi]\) are used for recording the \(m\)th path length, \(m = 2, \cdots, M\). To each undirected edge \((u, v)\) in \(G\), it constructs the directed edge in \(G^{M, \pi}\) from \((u, \zeta_1, \cdots, \zeta_M)\) to \((v, \lambda_1, \cdots, \lambda_M)\) and \(\lambda_m = \zeta_m + \varphi_m(u, v)\). Note that the infeasible paths of latter \(M - 1\) metrics have been filtered. So, an optimal path in \(G\) corresponds to an optimal path from \((s, 0, \cdots, 0)\) to \((d, \pi, \cdots, \pi)\).

Fig.1 illustrates a simple example of constructing the auxiliary graph when we set \(\pi_1 = 2, M = 2\). If we use the same topology of Fig.1 (a), and set \(\pi_1 = 3, M = 3\), another auxiliary graph will be constructed. Assuming that \(\varphi_2(s, a) = 1, \varphi_3(s, a) = 1, \varphi_2(a, d) = 1, \varphi_3(a, d) = 2\), thus, the shortest path computed by metric \(\varphi_1\) is \((s, 0, 0) \rightarrow (a, 1) \rightarrow (d, 2, 3) \rightarrow (d, 3, 3)\), note that the link between \((d, 2, 3)\) and \((d, 3, 3)\) is zero-length edge predefined in [14]. Hence, the corresponding path \(s \rightarrow a \rightarrow d\) in the original graph can be obtained.

The basic idea behind such methodology is using the space to trade for the time. As we know, Xue’s approximation scheme with time complexity of \(O(|E||\frac{1}{\epsilon}M - 1|)\) is currently the fastest one for path establishment in multi-constrained QoS routing. However, as we observed from the examples, the space complexity of Xue’s scheme is much considerable. It inspires us to seek an effective scheme to approximate the MCOP.

III. MAIN ALGORITHMS

The main part of the proposed algorithm, called CAGA (Constructing Auxiliary Graph Algorithm), is presented in Algorithm 1. It inherits the concept of constructing auxiliary graph. Instead of extending each node to large number of auxiliary nodes, we employ the nonlinear definition of path constraints [15] in the following to achieve the reduction of complexity.

\[
l_q(P) = \left( \sum_{m=1}^{M} \left( \frac{\varphi_m}{\Psi_m} \right)^q \right)^{\frac{1}{q}}
\]

Since the curved equilength line of Eq.(1) could match the constraint boundaries very well (see [15] Fig. 2) and the other
Algorithm 1 CAGA
1: Let $i = 1$
2: Extend $G(V, E)$ to a directed graph $G_{M,\pi_i}^i(V, E)$ with node set $V_{M,\pi_i}^i = V \times \{0, 1, 2, \ldots, \pi_i\}$ and edge set $E_{M,\pi_i}^i$.
To a given edge $(u, v)$ in $E$, $E_{M,\pi_i}^i$ contains directed edges from vertex $(u, \xi_i)$ to vertex $(v, \lambda_i)$ such that $\lambda_i = \xi_i + \lfloor \phi_i(u,v) \rfloor$, where $\phi_i(u,v) = \sum_{m=1}^{M-1} \left\lfloor \frac{\phi_m(u,v)}{\psi_m} \right\rfloor$; and directed edges from vertex $(v, \lambda_i)$ to vertex $(u, \xi_i)$ such that $\lambda_i = \xi_i + \lceil \phi_i(u,v) \rceil$. All such edges have the same length $\phi_i'(u,v)$, as well as $E_{M,\pi_i}^i$ contains zero-length edges from vertex $(u, \xi_i^0)$ to vertex $(v, \lambda_i^0)$ where $\lambda_i^0 = \xi_i^0 + 1$ if $\xi_i^0 < \pi_i$.
3: Calculate shortest paths $p_{M,\pi_i}^{i,\lambda_i^0}$ from the vertex $(s, 0)$ to vertices $(t, \lambda_i^0)$ in $G_{M,\pi_i}^i(V, E)$, $\lambda_i^0 = 1, \ldots, i - 1, i + 1, \ldots, \pi_i$.
if the length of path $p_{M,\pi_i}^{i,\lambda_i^0}$ is greater than $\pi_i$
\hspace{1cm} if $i < M$ then
\hspace{2cm} $i = i + 1$, goto step 2
\hspace{1cm} else
\hspace{2cm} Output No. STOP.
\hspace{1cm} end if
\hspace{1cm} end if
4: Find the smallest integer $\delta_i \leq \pi_i$ such that the shortest path $p_{M,\delta_i}^{i,\lambda_i^0}$ has length no more than $\delta_i$
5: if $i < M$ then
\hspace{1cm} $\pi_{i+1} = \delta_i$, $i = i + 1$ and goto step 2
else
\hspace{1cm} Output the path $p_W$ corresponding to $p_{M,\delta_M}^{i,\lambda_i^0}$ obtained by ignoring the latter component within each node along the path $p_{M,\delta_M}$.
end if

Algorithm 2 FPAS
1: Set $\pi_1 = \tau$
2: Apply CAGA in $G^\tau$, output the corresponding path $p_W$.

Since each $\varphi_m(e)$ is a positive real-value, for $m = 1, \ldots, i - 1, i + 1, \ldots, M$ and $e \in E$. The existence of directed edge from $(u, \xi_i)$ to $(v, \lambda_i)$ in $p_{M,\pi_i}^{i,\lambda_i^0}$ implies that $\xi_m < \lambda_m$ for $m = 1, \ldots, M$. Therefore, the graph $G_{M,\pi_i}^i(V, E)$ is acyclic. The worst case time complexity of each stage is $O(2^{|E|} \pi_i + |V| \pi_i) = O(|E| \pi_i)$. The algorithm runs at most $M$ times, and $\pi_{i+1} \leq \pi_i$, $i = 1, \ldots, M - 1$, thus, the time complexity can be rewritten as $O(\sum_{i=1}^{M} (2^{|E|} \pi_i + |V| \pi_i)) = O(M |E| \pi_1)$.

In practical settings, $M$ is a small value. As the network grows larger, more edges will be generated, but the size of constraints remains bounded. Thus, the size of constraints does not depend on the size of network, hence, we can assume that $M = O(1)$. So the worst case time complexity of CAGA is $O(|E| \pi_1)$.

Theorem 1 implies that the time complexity of CAGA is in the lower order as we achieve using the Xue based scheme, additionally, the CAGA may obtain more precisely path than Xue’s scheme after running $M$ times. In the following, we will present a quite simple FPAS based on CAGA.

For a given positive real number $\tau$, construct an auxiliary graph $G^\tau = (V, E, \varphi_1^\tau, \ldots, \varphi_M^\tau)$ which is the same as $G$ except that the edge weighting function $\varphi_m$ is changed to $\varphi_m^\tau$ such that $\varphi_m(e) \cdot \tau \leq \varphi_m^\tau(e) = \varphi_m(e) \cdot \tau + 1$ for every $e \in E$. Let $\tau = \frac{|V|-1}{\rho \psi_m}$, where $\psi \geq 1$, $\rho = 1$, we present the FPAS as follows, and base our analysis on Xue’s proof.

Theorem 2: Algorithm 2 FPAS finds a $(1 + \epsilon)$ approximation to MCOP.

Proof: In the $i$th stage, note that $G_{M,\pi_i}^i(V, E)$ has $|V| \pi_i + |V| \pi_i = O(|E| \pi_i)$ vertices and $O(2^{|E|} \pi_i + |V| \pi_i)$ edges (This confirms our observation that the space complexity is dramatically reduced). It conforms to the construction of $G_{M,\pi_i}^i(V, E)$ that $G_{M,\pi_i}^i$ has a directed path $p_{M,\pi_i}$ from $(u, \xi_i)$ to $(v, \lambda_i)$ with length $l$ if and only if the corresponding path $p_W$ in $G(V, E)$ (obtained by keeping only the first component in each vertex on $p_{M,\pi_i}$) satisfies $\varphi_m(p_W) = l$ and $\varphi_m(p_W) \leq \lambda_m - \xi_m$ for $m = 1, \ldots, i - 1, i + 1, \ldots, M$. Therefore, the path $p_W$ found in the Algorithm CAGA minimizes $\max_{1 \leq m \leq M} \varphi_m(p)$ among all paths in $G(V, E)$.

$M - 1$ metrics would not violate the constraints in the auxiliary graph, it is possible to use this way to aggregate $M - 1$ metrics into a single one.

From the steps of constructing the auxiliary graph, we observed that, the Algorithm 1 has much lower space complexity than Xue’s scheme by employing Eq. (1). The following theorem gives the time complexity of the Algorithm 1.

Theorem 1: Algorithm CAGA obtains a source to destination path $p_W$ which minimizes $\max_{1 \leq m \leq M} \varphi_m(p)$ among all paths in $G(V, E)$. The worst case time complexity of the algorithm is $O(|E| \pi_1)$.

Proof: In the $i$th stage, note that $G_{M,\pi_i}^i(V, E)$ has $|V| \pi_i + |V| \pi_i = O(|E| \pi_i)$ vertices and $O(2^{|E|} \pi_i + |V| \pi_i)$ edges (This confirms our observation that the space complexity is dramatically reduced). It conforms to the construction of $G_{M,\pi_i}^i(V, E)$ that $G_{M,\pi_i}^i$ has a directed path $p_{M,\pi_i}$ from $(u, \xi_i)$ to $(v, \lambda_i)$ with length $l$ if and only if the corresponding path $p_W$ in $G(V, E)$ (obtained by keeping only the first component in each vertex on $p_{M,\pi_i}$) satisfies $\varphi_m(p_W) = l$ and $\varphi_m(p_W) \leq \lambda_m - \xi_m$ for $m = 1, \ldots, i - 1, i + 1, \ldots, M$. Therefore, the path $p_W$ found in the Algorithm CAGA minimizes $\max_{1 \leq m \leq M} \varphi_m(p)$ among all paths in $G(V, E)$.

This implies that
$$\max_{1 \leq m \leq M} \varphi_m(p_{opt}) \leq \frac{|V|-1}{\rho \psi_m} + |V| - 1$$
(3)
According to the definition of $p_W$, we have
$$\max_{1 \leq m \leq M} \varphi_m(p_W) \leq \max_{1 \leq m \leq M} \varphi_m^\tau(p_{opt})$$
(4)
In terms of (3) and (4), we rewrite the inequality as
$$\max_{1 \leq m \leq M} \varphi_m(p_W) \leq \frac{|V|-1}{\rho \psi_m} + |V| - 1$$
(5)
On the other hand, in terms of definition of $\varphi_m^\tau(e)$ we have
$$\varphi_m^\tau(p_W) = \sum_{e \in p_W} \varphi_m^\tau(e) \geq \sum_{e \in p_W} \varphi_m(e) \cdot \frac{|V|-1}{\rho \psi_m} = \varphi_m(p_W) \cdot \frac{|V|-1}{\rho \psi_m}$$
(6)
Combining (5) and (6), by the definition of MCOP, we have

\[ \varphi_m(p^W) \cdot \frac{|V|-1}{p^W \cdot \varphi_m} \leq \eta \cdot \varphi_m \cdot \frac{|V|-1}{p^W \cdot \varphi_m} + |V| - 1 \]  

(7)

i.e.

\[ \varphi_m(p^W) \leq \rho \cdot \eta \cdot \varphi_m \cdot \epsilon \]
\[ \leq (1 + \epsilon) \cdot \rho \cdot \eta \cdot \varphi_m \]
\[ \leq (1 + \epsilon) \cdot \varphi_m \]  

(8)

Hence, \( p^W \) is a \((1 + \epsilon)\)-approximation to MCOP, the time complexity of FPAS is \( O\left(\frac{|E||V|}{\epsilon}\right) \).

Here we should identify some features of the proposed FPAS. For \( M = 2 \), FPAS has the same time complexity of Xue’s scheme. For \( M > 2 \), the proposed FPAS runs still in \( O\left(\frac{|E||V|}{\epsilon}\right) \) times which is faster than Xue’s scheme whose time complexity is \( O\left(|E|\left(\frac{|V|}{M}\right)^{M-1}\right) \). Moreover, by employing the nonlinear definition of path constraint, the presented FPAS has lower space complexity than Xue’s scheme. Apart from these, by introducing the limited iterations of the algorithm rather than rounding each metric into an integer, the proposed scheme could find a more precise solution. This means that the proposed FPAS could find a preferable \((1 + \epsilon)\)-approximation path with lower complexity performance.

IV. MARKOV ANALYSIS FOR PROPOSED FPAS

In general, the FPAS could obtain an optimal path that satisfies all the QoS requirements if there exists no background traffic in the network. However, if the background traffic has been in the network, the execution of FPAS would lead some “optimal” path to overload since all the optimal paths are computed in advance [16]. Such circumstance motivates us to reconsider the overall network performance based on the FPAS precomputation scheme.

We give the analytical model as shown in Fig. 2. Assuming the source node \( s \) can obtain \( n \) disjoint paths after the precomputation was performed, each path supports a single routing request at a time. Note that the source node should perform precomputation periodically with the arrival of routing requests, we assume the overall period of request arriving and possible “re-precomputation” follows the Poisson process with the arrival rate \( \alpha \). The completion of path selection is exponentially distributed with mean \( 1/\mu \).

We focus on the Markov property of analytical model in our analysis. Specifically, we are interested in the routing success probability \( Pr_s \), which is given by the ratio between the rate of the accepted requests \( \alpha_a \), and the rate of incoming requests \( \alpha \). Based on the above assumption, the Markov model can be depicted in Fig. 3.

To obtain \( Pr_s \), the system steady-state probability \( Pr_j, j = 1, \cdots, n \) should be found at first. The system flow-equations according to Fig. 3 is

\[ (\alpha + j\mu)Pr_j = (j + 1)\mu Pr_{j+1} \]  

(9)

then

\[ Pr_j = \frac{(\rho + j - 1)(\rho + j - 2) \cdots \rho}{j(j - 1) \cdots 1} Pr_0 \]
\[ = \left( \frac{\rho + j - 1}{j} \right)^n Pr_0 \]  

(10)

where \( \rho = \alpha/\mu \). Substituting (10) in the probability-conservation equation, that yields

\[ Pr_0 + Pr_0 \sum_{i=1}^{n} \left( \frac{\rho + j - 1}{j} \right) = Pr_0 \sum_{i=0}^{n} \left( \frac{\rho + j - 1}{j} \right) = 1 \]  

(11)

which yields a close-form solution for \( Pr_0 \) (and the other steady-state probabilities)

\[ Pr_0 = \frac{1}{\binom{\rho + n}{n}} \]  

(12)

Thus, we have the success probability \( Pr_s \) as

\[ Pr_s = 1 - Pr_n = \frac{n}{n + \rho} \]  

(13)

This equation means the routing success probability is decided by the total number of disjoint paths and system intensity \( \rho \).

V. SIMULATIONS AND PERFORMANCE EVALUATION

A. Simulation Results

In this part, we compare our FPAS to Xue’s one in terms of precision of solution and running time. The results were obtained on a IBM 2.4GHz PC with 2GB memory and running Linux. All running time reported are in seconds, and each of them is a mean of 20 independent tests. For our experiments we used the network topology generator [17] that all scenarios generated randomly as well as QoS parameters associated with each edge.
In the first scenario, we performed two FPASs on 1000, 2000, 3000, 4000 and 5000 nodes respectively under cost and delay constraints (Fig. 4). Fig. 4 refers that the proposed FPAS runs slower than Xue’s one, but remember what was mentioned in the preliminary section, the proposed FPAS can obtain a more precise path (we set $\epsilon = 20\%$ for Xue’s FPAS and $\epsilon = 10\%$ for our proposed one, in the next subsection, the performance of two paths will be examined). This situation naturally raises a question: if we set the same $\epsilon$ in both FPAS, which one runs faster? The next test will give the answer.

In the second scenario, we tested the running time of two FPASs with different $\epsilon$ ($\epsilon = 5\%, \epsilon = 10\%$) under three constraints. The scenarios are ranged from 200 to 2000 nodes and the results are shown in Fig. 5. Meanwhile, the confidential intervals of the results could be easily calculated by the results’ mean value and standard deviation. We do not give the confidential intervals for each test results herein due to space limitation. From the Fig. 5 we can see that, the proposed algorithm runs much faster than Xue’s FPAS with the same $\epsilon$, this follows the preliminary theoretical analysis that the proposed FPAS simplifies the executions by employing the nonlinear definition of path constraint; on the other hand, we predict that, there exists a $5\% < \epsilon_0 < 10\%$ for proposed FPAS that makes the running time of both FPASs equal. In sum, the above two properties imply that the proposed FPAS can run faster than Xue’s one if the founded paths have the same precision, and our proposed FPAS is capable of obtaining a more precise solution than Xue’s scheme if the running time of two FPAS are the same. In other words, the proposed FPAS has much better performance.

**B. Performance Evaluation**

The main purpose of this part is to test the performance of two paths obtained by Xue’s scheme (denoted by Path1 in the figures) and our presented one (denoted by Path2). Both paths are obtained with the same running time. The simulation scenario is shown in Fig. 6. In particular, we focus on the quantitative aspects of performance including delivery delay and throughput. The evaluations are conducted by OPNET Modeler 14.0 Education Version, and we configure the database-access service for each path, service parameters are typically set as shown in Table I.

Fig. 7 shows the performance of two paths obtained among 1000 nodes, the curves imply that the path which is computed by the presented scheme has lower delivery delay and higher throughput than those of Xue’s. This conforms to the above simulation result. Fig. 8 is the performance of two paths obtained among 5000 nodes, it also infers the same conclusion as Fig. 7 shows.

**VI. Conclusion**

In this paper, we revisit the problem of multiconstrained optimal path with $M$ constraints. We have presented an effective fully polynomial approximation scheme (FPAS) based on the technique of auxiliary graph construction. The FPAS used the nonlinear definition of the path constraint and limited iterations of the algorithm to achieve the low complexity and to obtain the precise path, respectively. We have further theoretically analyzed the Markov properties of the entire network on

<table>
<thead>
<tr>
<th>Attribute</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Mix (Queries/Total Transactions)</td>
<td>100%</td>
</tr>
<tr>
<td>Transaction Interarrival Time</td>
<td>exponential (12)</td>
</tr>
<tr>
<td>Transaction Size</td>
<td>constant (512)</td>
</tr>
<tr>
<td>Symbolic Server Name</td>
<td>Database Server</td>
</tr>
<tr>
<td>Type of Service</td>
<td>Best Effort (0)</td>
</tr>
<tr>
<td>RSVP Parameters</td>
<td>None</td>
</tr>
</tbody>
</table>
the basis of FPAS precomputation scheme. Some important parameters reflecting routing success probability are obtained. We have also experimented on different scale of random networks and compared our method and results with previous well known studies. Our study shows that our proposed FPAS is efficient in both running time and precision of the solution.

As future research, we plan to study the QoS routing in wireless networks and propose a wireless version of FPAS based on our current research.

ACKNOWLEDGMENT

The authors would like to thank Dr. S. Chen and Dr. D.H. Lorenz for their suggestive comments to this paper. This work is supported by the National Basic Research Program (973 Program) of China, under grant No. 2007CB310701.

REFERENCES