QoS Routing with Topology Aggregation in Networks

Ronghui Hou, King-Shan Lui, Ka-Cheong Leung, and Fred Baker

Abstract

In this paper, we propose a novel method for aggregating the QoS information with two independent additive metrics. To the best of our knowledge, our approach is the first study to use the *area-minimization optimization*, the de facto optimization problem of the QoS information aggregation. In our approach, a set of real numbers with constant size is used to approximate the supported QoS between different domains, so that the advertisement overhead and the space requirement is independent of the network size and topology, which is scalable. The simulation results show that the proposed method outperforms the existing methods.

Index Terms

QoS routing, hierarchical networks, additive constraints, topology aggregation.

I. INTRODUCTION

Supporting Quality-of-Service (QoS) in the Internet is a big challenge due to the scalability problem. In the Internet, nodes are hierarchically grouped into different domains. Each node in a domain has no topology information of other domains. For computing the supported QoS between different domains, each border has to advertise the supported QoS information from itself to a destination to its neighbors, so that the neighbors can obtain the supported QoS to the same destination. We use the following example to illustrate this problem.

Fig. 1 shows an Internet topology which contains five domains and each domain contains two border nodes. Each border node is connected with other border nodes in other domains or in the same domain. The border $X.i$ means the border i in domain X. Now, we consider the process of computing the supported

R. Hou, K.-S. Lui, and K.-C. Leung are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong (e-mail: {rhhou, kslui, kcleung}@eee.hku.hk).

Fred Baker is with Cisco Research Center, San Jose, CA 95134, USA (e-mail: fred@cisco.com).

This work is supported in part by the Cisco Research Initiative Award.

QoS from $S.2$ to the domain T. We can see that $B.2$ and $C.2$ are directly connected with domain T, so, they advertise the supported QoS to $B.1$ and $C.1$, respectively. Secondly, $B.1$ and $C.1$ compute the supported QoS from themselves to domain T and advertise them to $A.2$ and $S.2$, respectively. The process continues until $S.2$ receives the supported QoS from $A.1$ to T and that from $C.1$ to T. Based on the received QoS information, $S.2$ can compute the supported QoS from itself to domain T. However, the size of the supported QoS information depends on the network size and topology, as discussed as follows.

In this work, we consider two independent additive constraints, the QoS parameter of link l is denoted by (c_l, d_l), where c_l and d_l are the cost metric and delay metric of link l. For a path $p = \{l_1, l_2, \ldots, l_h\}$, the QoS parameter of p is (c_p, d_p), where $c_p = \sum_{i=1}^{h} c_{l_i}$ and $d_p = \sum_{i=1}^{h} d_{l_i}$. Each QoS parameter corresponds to a point on the cost-delay plane. For two different QoS parameters, that one has smaller cost metric and the other has a smaller delay constraint, we cannot say that one is better than the other. We call these QoS parameters as representative points. For example, in Fig. 2, the points $\{p_1, p_3, p_5, p_7\}$ are representative points, and the shaded area is the feasible region defined by these representative points. For any request requirement falling in this area, we can find at least one physical path satisfying this requirement. Therefore, the supported QoS is defined by a set of representative points. However, advertising all these representative points is not scalable. To solve the scalability problem, the supported QoS, which is defined by a staircase, has to be aggregated. A scalable QoS information aggregation method is thus necessary for supporting QoS in the Internet [16].

There have been a few works on the QoS information aggregation. Generally, the works in [1], [3], [10], [13] just consider one QoS parameter aggregation. The logical link between any two borders is selected as the “best” path by using the shortest-path algorithm. The work in [4] considers multiple QoS parameter. Each QoS parameter of the logical link between any two borders is set to be the best one, which is found by using the shortest-path algorithm with respective to the given metric. The work in [18] study the advertisement of the QoS information between different domains under the condition of the inaccurate link state. It proposed a statistic composite to represent a single metric of the path between any two borders. The work in [5] uses a curve on the cost-delay plane to approximate the staircase. However, the work gives a precedence to one of the metrics and the information about the other metric is lost, moreover, it did not provide a polynomial-time routing algorithm corresponding to this method. The work in [8] was the first to propose a method to aggregate the supported QoS and the corresponding polynomial-time routing algorithm. In [8], a line segment is used to approximate the staircase, as illustrated in Fig. 2. The

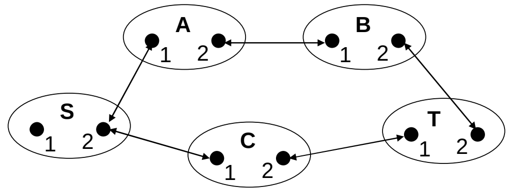


Fig. 1. A simple network.

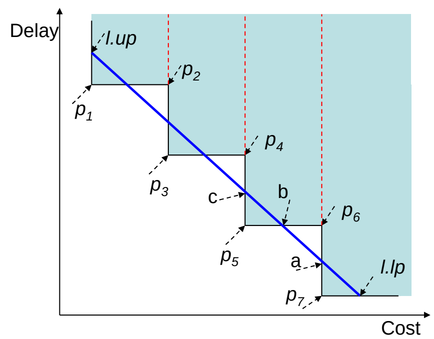


Fig. 2. Line segment approximation scheme.

works in [6], [14] studies the distance-vector routing problem with additive-concave and concave-concave metrics based on the line segment aggregation method. However, the line segment in [8] is computed by using the method of Least Square, which is not the actual optimization problem of the QoS information aggregation.

In this paper, we propose a new way of representing the aggregated state by a set of representative points, which has a constant size, so that it is scalable for advertising the aggregated supported QoS. To the best of our knowledge, we are is the first to study the *area-minimization optimization*, the de facto optimization problem of the QoS information aggregation. We evaluate the performance of our algorithm, and the simulation results show that our algorithm outperforms the existing algorithm due to smaller distortion and smaller advertisement overhead.

II. PROBLEM FORMULATION

As mentioned in the previous section, the supported QoS is defined by a set of representative points which form a staircase. A line segment was proposed in [8] to approximate this staircase. Since line segment is just an approximation of the supported QoS, it may introduce some distortion. For example, the line segment may overestimate the minimum delay value, so that some feasible regions may not be included in the approximated feasible region, such as the triangle Δbcp_3 in Fig. 2. We call such regions as *overestimated area*, denoted by $\Delta_+(l)$. While for some cost constraints, the line segment may underestimate the minimum delay value, so that some infeasible regions may be induced by the line segment, such as the triangle Δabp_2 in Fig. 2. We call such regions as *underestimated area*, denoted by $\Delta_-(l)$. In order to minimize the approximation error, the line segment should be computed such that the sum of the *overestimated areas* and *underestimated areas* is minimized, and we call such optimization as area-minimization optimization. Although the works in [14], [15] mentioned the area-minimization optimization, they did not give any solution.

Before discussing the area-minimization, we would like to give the overview of the line segment approximation method in the following section.

III. LINE SEGMENT APPROXIMATION

In [8], a line segment topology aggregation scheme was proposed to approximate the region of the supported service. The details on the line segment aggregation method can be referred to [2], [8], we just give an overview about this method in this section. As illustrated in Fig. 2, the shaded area is the optimal feasible region defined by a staircase. We use a line segment to approximate the staircase, the upper right area on the line segment is the approximated feasible region. The line segment l can be defined by two points $[l.up, l.lp]$, where $l.up$ and $l.lp$ are the upper end point and lower end point of l, respectively. Given a point p which corresponds to a specific QoS parameter, denote $p.c$ and $p.d$ as the cost value and the delay value of p, respectively.

In order for the routing protocol applies the line segment aggregation method, we need to consider the join operation of the supported QoS. We give the problem statement as follows.

Suppose that border bs has a neighbor j which computes the supported QoS from itself to border bd. Assume that bs has computed the supported QoS from itself to j, we need to consider how to compute the supported QoS from bs to bd, via j. We refer this operation as the *join operation* and is denoted by $\oplus$.

If the two QoSes are represented by the points p_1 and p_2, respectively, $p_1 \oplus p_2 = (p_1.c + p_2.c, p_1.d + p_2.d)$. We formally prove that $p \oplus l = [p \oplus l.up, p \oplus l.lp]$. If the supported QoS from bs to j is represented by a point p and that from j to bd is represented by a line segment l, the supported QoS from bs to bd, via j, is represented by a new line segment, which is obtained by shifting l horizontally by $p.c$ units to left and then shifting vertically the new line by $p.d$ units upwards on the cost-delay plane. The join operation is illustrated in Fig. 3.

If the two QoSes are both represented by two line segment l_1 and l_2, the result of $l_1 \oplus l_2$ may be composed by two line segments. Assume that l_2 is steeper than l_1, we formally prove that $l_1 \oplus l_2$ is

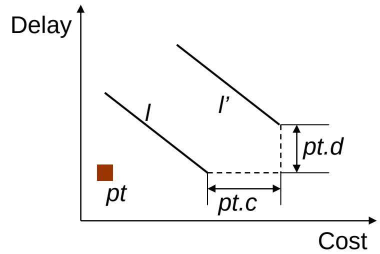


Fig. 3. An illustration of joining of a point and a line segment.

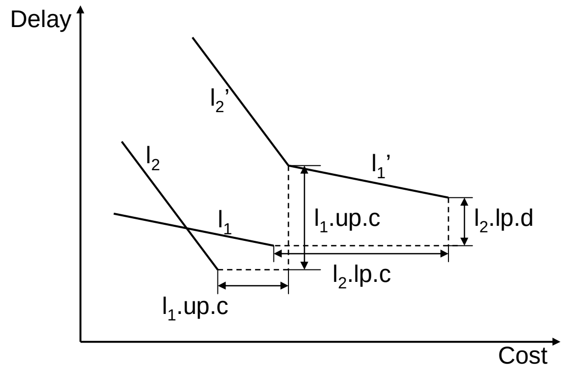


Fig. 4. An illustration of joining two line segments.

composed by two line segments $l_1.up \oplus l_2$ and $l_2.lp \oplus l_1$. The joining results of l_1 and l_2 is illustrated in Fig. 4.

If border bs has multiple border neighbors, different neighbors provide different supported QoSes to a destination bd. Therefore, the total supported QoS from bs to bd is defined by multiple line segments. As illustrated in Fig. 5, bs has three neighbors which provide the supported QoSes defined by l_1, l_2, and l_3, respectively. The shaded area is the total supported QoS from bs to bd, which is defined by a polyline. We call the polyline as the service outline. After bs obtains the service outline, it will find a line segment approximating the polyline, such as the line segment l in Fig. 5. The details about the line segment approximation method can be found in [2].

IV. AREA-MINIMIZATION BASED LINE SEGMENT

Based on the discussion in the previous section, line segment is computed by approximating a staircase or a polyline. We call either staircase or polyline the service outline. In [8], line segment is computed by using linear regression. For instance, assume that the line segment is represented by the linear function $y = m \cdot x + b$, where m and b are the slope and the y-coordinate intercept of the line segment, respectively. In Fig. 2, the line segment is computed such that $\sum_{i=1}^{7}(p_i.d - (m \cdot p_i.c + b))^2$ is minimized. In Fig. 5, the line segment l is computed such that $\sum_{i=1}^{5}(p_i.d - (m \cdot p_i.c + b))^2$. We can see that linear regression is

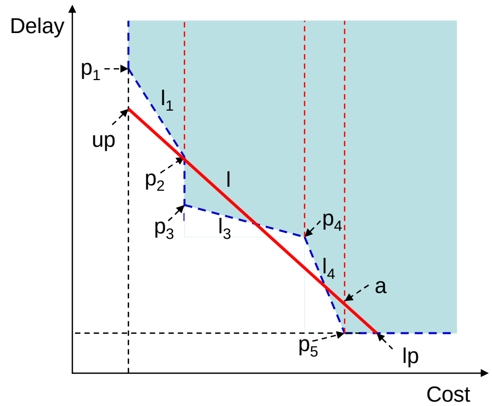


Fig. 5. Illustration for the total supported QoS.

not an area-minimization optimization. The line segment computed by linear regression is not an optimal approximated line segment, while the optimal approximated line segment should be computed such that the sum of the underestimation area and the overestimation area is minimized. Accordingly, we would like to discuss how to compute an approximated line segment based on area-minimization optimization.

The service outline is defined by a set of points called service outline points. For example, in Fig. 2, the service outline points are $\{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$. In Fig. 5, the service outline points are $\{p_1, p_2, p_3, p_4, p_5\}$. We can also say that the feasible region is defined by the service outline points.

Assume that the real feasible region is defined by the service outline points $\{p_1, \ldots, p_n\}$, where $p_{i+1}.c \geq p_i.c$ for all $i = 1, \ldots, n-1$. We divide the cost coordinate into several ranges $\{[p_1.c, p_2.c], [p_2.c, p_3.c], \ldots, [p_{n-1}.c, p_n.c], [p_n.c, \infty)\}$. The feasible region included in the subregion $[p_i.c, p_{i+1}.c] \times [0, \infty)$ is defined by the line segment $[p_{i+1}, p_i]$. For example, in Fig. 2, the feasible region included in $[p_1.c, p_2.c] \times [0, \infty)$ is defined by the line segment $[p_1, p_2]$. In Fig. 5, the feasible region included in $[p_3.c, p_4.c] \times [0, \infty)$ is defined by the line segment $[p_3, p_4]$. The feasible region in the region $[p_n.c, \infty) \times [0, \infty)$ is defined by the point p_n. For instance, in Fig. 5, p_5 defines the feasible region include in $[p_5.c, \infty) \times [0, \infty)$. It is obvious, if $p_i.c = p_{i+1}.c$, the feasible region included in $[p_i.c, p_{i+1}.c] \times [0, \infty)$ is $\varnothing$. We consider that the feasible region defined by a vertical line is $\varnothing$.

Given an approximated line segment l, we are going to discuss the approximation error corresponding to the subregion $\mathbf{S}_i = [p_i.c, p_{i+1}.c] \times [0, \infty)$, where $i = 1, \ldots, n-1$. Assume that l can be defined by a linear function $y = m \cdot x + b$, where m and b are the slope and the y-coordinate intercept of l, respectively. For instance, in Fig. 6(b), the bold solid line is the line segment $[p_i, p_{i+1}]$ and the bold dashed line is the approximated line segment l, the yellow areas denote the approximation error produced by l in $\mathbf{S}_i$. For computing the approximation error in $\mathbf{S}_i$, we need to consider four cases:

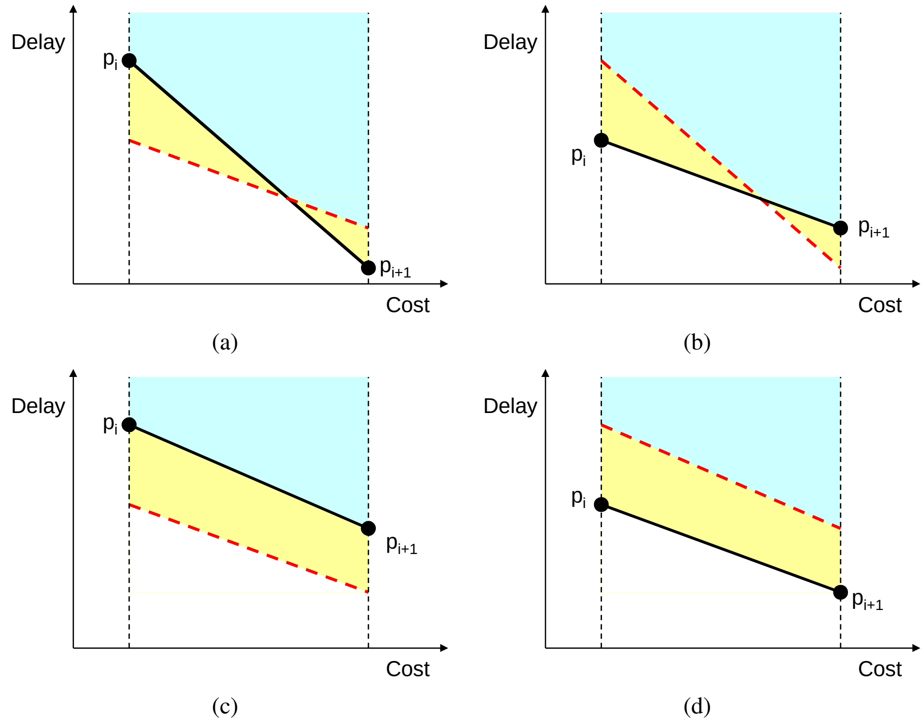


Fig. 6. An example illustrating the computation of line segment based on area-minimization.

1) If $m \cdot p_i.c + b < p_i.d$ and $m \cdot p_{i+1}.c + b > p_{i+1}.d$, as illustrated in Fig. 6(a).
2) If $m \cdot p_i.c + b > p_i.d$ and $m \cdot p_{i+1}.c + b < p_{i+1}.d$, as illustrated in Fig. 6(b).
3) If $m \cdot p_i.c + b \leq p_i.d$ and $m \cdot p_{i+1}.c + b \leq p_{i+1}.d$, as illustrated in Fig. 6(c).
4) If $m \cdot p_i.c + b \geq p_i.d$ and $m \cdot p_{i+1}.c + b \geq p_{i+1}.d$, as illustrated in Fig. 6(d).

Let m_i and b_i be the slope of the line segment $l_i = [p_i, p_{i+1}]$. For any point p on l_i, it holds that $p.d = m_i \cdot p.c + b_i$. In Case I, there exists an intersection p between l_i and l, where $p.c = \frac{b_i - b}{m - m_i}$. We thus compute the underestimated area produced by l in $\mathbf{S}_i$, denoted by $\Delta_{-,i}(l)$ as follows.

$$\begin{aligned} \Delta_{-,i}(l) &= \int_{p_i.c}^{\frac{b_i-b}{m-m_i}} (m_i x + b_i) - (mx + b)dx \\ &= \tfrac{(b-b_i)^2}{2(m-m_i)} - (\tfrac{m_i - m}{2} p_i.c^2 + (b_i - b)p_i.c) \end{aligned} \tag{1}$$

The overestimated area, denoted by $\Delta_{+,i}(l)$, is as follows.

$$\begin{aligned} \Delta_{+,i}(l) &= \int_{\frac{b_i-b}{m-m_i}}^{p_{i+1}.c} (mx + b) - (m_i x + b_i)dx \\ &= \tfrac{(b-b_i)^2}{2(m-m_i)} - (\tfrac{m_i - m}{2} p_{i+1}.c^2 + (b_i - b)p_{i+1}.c) \end{aligned} \tag{2}$$

Therefore, Under the condition of Case I, the approximation error in the region $\mathbf{S}_i$ produced by l is:

$$\begin{aligned} E^1_{p_i,p_{i+1}} &= \Delta_{-,i}(l) + \Delta_{+,i}(l) \\ &= \tfrac{(b-b_i)^2}{(m-m_i)} - \Big(\tfrac{1}{2}(m_i - m)(p_i.c^2 + p_{i+1}.c^2) \\ &\quad + (b_i - b)(p_i.c + p_{i+1}.c)\Big) \end{aligned} \tag{3}$$

Denote $E^k_{p_i,p_{i+1}}$ as the approximation error in the region $\mathbf{S}_i = [p_i.c, p_{i+1}.c] \times [0, \infty)$ produced by the approximated line segment l in the Case k, where the feasible region in $\mathbf{S}_i$ is defined by the line segment $[p_i, p_{i+1}]$. We can easily verify that $E^2_{p_i,p_{i+1}} = E^1_{p_i,p_{i+1}}$.

In Case III, the approximation error is composed by the underestimated area, as illustrated in Fig. 6(c). We have

$$E^3_{p_i,p_{i+1}} = \frac{1}{2}(m_i - m)(p_{i+1}.c^2 - p_i.c^2) + (b_i - b)(p_{i+1}.c - p_i.c). \tag{4}$$

We also have $E^4_{p_i,p_{i+1}} = -E^3_{p_i,p_{i+1}}$. In Case IV, the approximation error is composed by the overestimated area.

Given the real supported QoS defined by the service outline points $\{p_1, \ldots, p_n\}$, where p_1 and p_n are the minimum cost and minimum delay points, respectively. The upper end point and the lower end point of the corresponding approximated line segment l should have the cost value of $p_1.c$ and the delay value of $p_n.d$, respectively. For example, in Fig. 2, $l.up.c = p_1.c$ and $l.lp.d = p_7.d$. In Fig. 5, $up.c = p_1.c$ and $lp.d = p_5.d$.

We have discussed how to compute the approximation error in the region $\mathbf{S}_i = [p_i.c, p_{i+1}.c] \times [0, \infty)$ for all $i = 1, \ldots, n-2$. Now, we discuss how to compute the approximation error included in the region $[p_{n-1}.c, \infty) \times [0, \infty)$. We need to consider two cases:

1) If $m \cdot p_n.c + b \geq p_n.d$.
2) If $m \cdot p_n.c + b < p_n.d$.

In Case I, the lower end point of the approximated line segment has cost value no less than the cost value of the minimum delay point p_n, as illustrated in Fig. 2 and Fig. 5. We compute the approximation error in the region $\mathbf{S}_{n-1} = [p_{n-1}.c, p_n.c] \times [0, \infty)$ and the region $\mathbf{S}_n = [p_n.c, \infty) \times [0, \infty)$, separately. Since $m \cdot p_n.c + b \geq p_n.d$, the approximation error in $\mathbf{S}_{n-1}$ can be represented by $E^1_{p_{n-1},p_n}$ or $E^3_{p_{n-1}.c,p_n.c}$, which depends on the relationship of $m \cdot p_{n-1}.c + b$ and $p_{n-1}.d$. The approximation error in the region

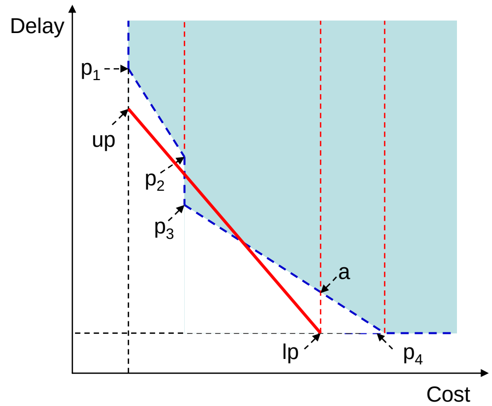


Fig. 7. Illustration for the approximated line segment.

$\mathbf{S}_n$ is composed by the overestimated area, which is a right triangle. For example, in Fig. 2, the triangle $\Delta ap_7 l.lp$ is the approximation error in the region $[p_7.c, \infty) \times [0, \infty)$. In Fig. 5, the triangle $\Delta ap_5 lp$ is the approximation error in the region $[p_5.c, \infty) \times [0, \infty)$. We can see that the approximation error in $\mathbf{S}_n$ can be represented by $E^4_{p_n.c,lp}$, where $lp.c = \frac{p_n.d-b}{m}$ is the cost value of the lower end point of l.

In Case II, the lower end point of the approximated line segment has the cost value smaller than that of the minimum delay point p_n. In Fig. 7, $lp.c < p_4.c$. In this case, the feasible region defined by l includes the feasible region in the region $\mathbf{S}_n = [p_n.c, \infty) \times [p_n.d, \infty)$, moreover, lp has the same delay value as p_n, so, l will not induce the infeasible region in $\mathbf{S}_n$. Therefore, the approximation error in $\mathbf{S}_n$ is $\varnothing$. We thus need to compute the approximation error in the region $\mathbf{S}_{n-1} = [p_{n-1}.c, p_n.c] \times [0, \infty)$. We divide $\mathbf{S}_{n-1}$ into two subregions $\mathbf{S}^1_{n-1} = [p_{n-1}.c, lp.c] \times [0, \infty)$ and $\mathbf{S}^2_{n-1} = [lp.c, p_n.c] \times [0, \infty)$. The approximation error in $\mathbf{S}^1_{n-1}$ can be represented by $E^2_{p_{n-1},lp}$ or $E^4_{p_{n-1},lp}$, which depends on the relationship of $m \cdot p_{n-1}.c + b$ and $p_{n-1}.d$. The approximation error in the region $\mathbf{S}^2_{n-1}$ can be represented by E^3_{lp,p_n}.

Suppose that a set of service outline points $\{p_1, \ldots, p_n\}$ and an approximated line segment l which is defined by $y = mx + b$. For each point p_i, where $i = 1, \ldots, n$, we need to first assume whether $m \cdot p_i.c + b$ is larger than $p_i.d$ or not. Based on the assumption, we can compute the approximation error produced by l, and then we compute a local optimal approximated line segment with the given constraints. There are totally 2^n different cases, so, we need to compute 2^n different local approximated line segments. Finally, we select the one with the minimum approximation error as the global approximated line segment.

Now, we discuss the characteristics of the formulation of the approximation error produced by l with a set of given constraints. Let the line segment l be represented by $y = mx + b$, and the service outline l_i be represented by $y = m_i x + b_i$ for all $i = 1, \ldots, n-1$. For instance, in Fig. 5, we first give a set of constraints as follows.

1) $m \cdot p_1.c + b \leq p_1.d$

2) $m \cdot p_2.c + b \leq p_2.d$

3) $m \cdot p_3.c + b \geq p_3.d$

4) $m \cdot p_4.c + b \leq p_4.d$

5) $m \cdot p_5.c + b \geq p_5.d$

Based on the above constraints, we can compute the approximation error as $\mathbf{E} = E^3_{p_1,p_2} + E^1_{p_2,p_3} + E^2_{p_4,p_5} + E^4_{p_5,lp}$.

The formulation of the approximation error produced by a line segment l with a given set of constraints can be represented by the following way:

$$f_e = \sum_{i=1}^{k_1} \frac{(b-b_i)^2}{m-m_i} + \sum_{i=k_1+1}^{k_2} \frac{(b-b_i)^2}{m_i-m} + Pm + Qb \tag{5}$$

Each line segment l_i, which is represented by $y = m_i x + b$, where $i = 1, \ldots, k_1$, is steeper than the approximated line segment l, since the approximation error is represented by (3). Each line segment l_i, which is represented by $y = m_i x + b$, where $i = k_1+1, \ldots, k_2$, is flatter than the approximated line segment l, since the approximation error is represented by negative ((3)). Therefore, for each $i = 1, \ldots, k_1$, they hold that $m - m_i > 0$ and $b < b_i$. For each $i = k_1+1, \ldots, k_2$, they hold that $m_i - m > 0$ and $b > b_i$.

We then need to find the optimal values of m and b so that f_e is minimized. As we know, if f_e is a non-convex, computing the optimal values of m and b is NP-hard. We thus need to first analyze the characteristics of f_e. Therefore, we compute the Hessian matrix of f_e, $\mathbf{H}(f_e)$ as follows.

$$\begin{pmatrix} \frac{\partial^2 f_e}{\partial^2 m} & \frac{\partial^2 f_e}{\partial m \partial b} \\ \frac{\partial^2 f_e}{\partial b \partial m} & \frac{\partial^2 f_e}{\partial^2 b} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{k_1} \frac{(b-b_i)^2}{(m-m_i)^3} + \sum_{i=k_1+1}^{k_2} \frac{(b-b_i)^2}{(m_i-m)^3} & \sum_{i=1}^{k_1} \frac{-2(b-b_i)}{(m-m_i)^2} + \sum_{i=k_1+1}^{k_2} \frac{2(b-b_i)}{(m_i-m)^2} \\ \sum_{i=1}^{k_1} \frac{-2(b-b_i)}{(m-m_i)^2} + \sum_{i=k_1+1}^{k_2} \frac{2(b-b_i)}{(m_i-m)^2} & \sum_{i=1}^{k_1} \frac{2}{m-m_i} + \sum_{i=k_1+1}^{k_2} \frac{2}{m_i-m} \end{pmatrix} \tag{6}$$

Each element of $\mathbf{H}(f_e)$ is positive. Let

$$\mathbf{H}(f_e) = \begin{pmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{B} & \mathcal{C} \end{pmatrix}. \tag{7}$$

Therefore $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ are all positive. Now, we compute the eigenvalues of $\mathbf{H}(f_e)$ are

$$\lambda = \frac{(\mathcal{A}+\mathcal{C}) \pm \sqrt{(\mathcal{A}-\mathcal{C})^2 + 4\mathcal{B}^2}}{2}. \tag{8}$$

Therefore, the two eigenvalues of $\mathbf{H}(f_e)$ are both positive, that means f_e is convex. Theoretically, we can compute the optimal values of m and b. However, in order to obtain the optimal values of m and b, we need to solve the following equations.

$$\begin{cases} \frac{\partial f_e}{\partial m} = \sum_{i=1}^{k_1} \frac{-(b-b_i)^2}{(m-m_i)^2} + \sum_{i=k_1+1}^{k_2} \frac{(b-b_i)^2}{(m_i-m)^2} + P = 0 \\ \frac{\partial f_e}{\partial b} = \sum_{i=1}^{k_1} \frac{2(b-b_i)}{m-m_i} + \sum_{i=k_1+1}^{k_2} \frac{2(b-b_i)}{m_i-m} + Q = 0 \end{cases} \tag{9}$$

As k_2 increases, it is very difficult to obtain the solution of (9). Moreover, for computing an optimal approximated line segment, we need to compute 2^n local optimal line segments and then select the one with the minimum error, where n is the number of the service outline points. n is proportional to the number of the border neighbors. We can see that computing an approximated line segment that minimizes the approximation error is very complicated. We prefer to applying a staircase to approximate the supported QoS. We will discuss our approach in the following section.

V. THE METHOD OF APPROXIMATE STAIRCASE

We now describe how to find an approximate staircase with minimized error in details. To facilitate our discussion, we first define the concept of *representativeness*, which is also defined in [2].

Definition 1: An QoS parameter p is more representative than another QoS parameter p' if and only if

1) $p.c \neq p'.c$ or $p.d \neq p'.d$, and;
2) $p.c \leq p'.c$ and $p.d \leq p.d$.

Given two points p_1 and p_2, if p_1 is more representative than p_2, the feasible region provided by p_2 is included in that by p_1.

Lemma 1: Given the real supported QoS defined by $\mathcal{RP}$ and the approximated staircase $\mathcal{AP}$. Each point in $\mathcal{AP}$ must be a point in $\mathcal{RP}$ or be located in the infeasible region spanned by $[c_1, c_n] \times [d_n, d_1]$, where (c_1, d_n) and (c_n, d_1) are the minimum-cost and minimum-delay points in $\mathcal{RP}$, respectively.

Proof: Assume that an approximated point $ap = (x, y)$ in $\mathcal{AP}$ is located in the feasible region and it is not a point in $\mathcal{RP}$. We thus can find a representative point rp in $\mathcal{RP}$ which is more representative than ap. rp does not induce infeasible region, but defines more feasible region than ap. Therefore, If we replace ap with rp, the approximation error produced by $\mathcal{AP}$ will be reduced.

The lemma is thus proved. ■

For example, The shaded area in Fig. 8 is the infeasible region included in the region $[c_1, c_5] \times [d_5, d_1]$. The approximated points must be located in this area. For the simplicity, we first consider the case of

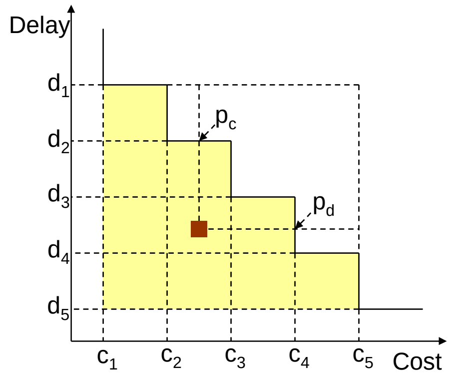


Fig. 8. An illustration for staircase approximation.

$\mathcal{H} = 3$. Since ap_1 and ap_3 are the minimum-cost and minimum-delay QoS parameters, respectively, we will discuss how to compute ap_2 in the following section.

A. Three points approximation

In this case, there is just one approximated representative point. Denote the approximated point as $ap_2 = (x, y)$, the *overestimated area* caused by $\mathcal{AP}$ as $\Delta_+(\mathcal{AP})$, and the *underestimated area* caused by $\mathcal{AP}$ as $\Delta_-(\mathcal{AP})$. We thus compute the optimal values of x and y such that $\Delta_+(\mathcal{AP}) + \Delta_-(\mathcal{AP})$ is minimized. Since x and y are unknown variables, it is impossible to obtain a general formulation representing the approximation error caused by $\mathcal{AP}$. We thus use the idea of *branch-and-bound*.

We mentioned earlier that the approximation error caused by $\mathcal{AP}$ must be included in the region $[c_1, c_n] \times [d_n, d_1]$. For simplicity, given a point p in $\mathcal{RP}$ or $\mathcal{AP}$, which is not rp_1 or rp_n, we can consider that the feasible region defined by p is $[p.c, c_n] \times [p.d, d_1]$.

Define the interval set $\mathbb{S}_c = \{[c_1, c_2], \ldots, [c_{n-1}, c_n]\}$ and the set $\mathbb{S}_d = \{[d_n, d_{n-1}], \ldots, [d_2, d_1]\}$. We thus obtain a set of regions $\mathbf{S} = \mathbb{S}_c \times \mathbb{S}_d$. Given any region $\mathbf{S}_{i,j} = [c_i, c_{i+1}] \times [d_{j+1}, d_j]$ in $\mathbf{S}$, we compute a local optimal point in $\mathbf{S}_{i,j}$, denoted by $ap_{i,j}$ for all $i, j = 1, 2, \ldots, n-1$. Finally, we select the global optimal point from all these local optimal points. Note that by Lemma 1, we do not need to consider $\mathbf{S}_{i,j}$ which is located in the real feasible region.

For example, in Fig. 8, we will compute the local optimal points corresponding to the ten regions $\{\mathbf{S}_{1,1}, \mathbf{S}_{1,2}, \mathbf{S}_{1,3}, \mathbf{S}_{1,4}, \mathbf{S}_{2,2}, \mathbf{S}_{2,3}, \mathbf{S}_{2,4}, \mathbf{S}_{3,3}, \mathbf{S}_{3,4}, \mathbf{S}_{4,4}\}$. In order to reduce the computational overhead, we introduce the following lemma.

Lemma 2: The global optimal point must not be located in the regions $\mathbf{S}_{1,j}$ and $\mathbf{S}_{i,n-1}$ for all $i, j = 1, 2, \ldots, n-1$.

Proof: We first consider the region $\mathbf{S}_{1,j} = [c_1, c_2] \times [d_{j+1}, d_j]$, where $j = 1, 2, \ldots, n-1$. Given any point $p_1 = (c, d)$ in $[c_1, c_2] \times [d_{j+1}, d_j]$, let $p_2 = (c_2, d)$. Define the feasible region defined by p_1 as R_1 and that by p_2 as R_2. We know that $R_1 \setminus R_2 = [c_1, c_2] \times [d, d_1]$. However, the region $[c, c_2] \times [d, d_1]$ must be the infeasible region. Therefore, the approximation error induced by p_2 is smaller than that by p_1. For the same reason, given any point $p_1' = (c, d)$ in the region $[c_i, c_{i+1}] \times [d_n, d_{n-1}]$, we can find the point $p_2' = (c, d_{n-1})$ such that the approximation error of p_2' is smaller than that of p_1'.

The lemma is thus proved. ■

Lemma 3: If the region $\mathbf{S}_{i,j} = [c_i, c_{i+1}] \times [d_{j+1}, d_j]$ is included in the infeasible region, we have $i \leq j$.

Proof: The point (c_i, d_{j+1}) is more representative than any point in the region $[c_i, c_{i+1}] \times [d_{j+1}, d_j]$. Assume that $j < i$, the point (c_i, d_i) is more representative than or equal to (c_i, d_{j+1}). In that case, the region $[c_i, c_{i+1}] \times [d_{j+1}, d_j]$ is included in the feasible region, which contradicts our assumption. ■

By Lemma 2 and Lemma 3, we do not have to compute the local optimal points in the regions $\mathbf{S}_{1,j}$ and $\mathbf{S}_{i,n-1}$ for all $i, j = 1, 2, \ldots, n-1$. That is to say, we just need to compute the local optimal point in the region $\mathbf{S}_{i,j}$ for all $i, j = 2, 3, \ldots, n-2$ such that $i \leq j$. For example, in Fig. 8, we just need to consider the regions $\{\mathbf{S}_{2,2}, \mathbf{S}_{2,3}, \mathbf{S}_{3,3}\}$.

Define the function $f_o(x, y)$ as the sum of the overestimated region and the underestimated region produced by the point (x, y). We thus obtain a typical *nonlinear programming problem (NLP)* as expressed as follows:

$$
\begin{aligned}
&\min f_o(x, y) \\
&\text{s.t.} \quad c_i \leq x \leq c_{i+1} \\
&\qquad\quad d_{j+1} \leq y \leq d_j
\end{aligned}
\tag{10}
$$

Now, we discuss how to compute objective function $f_o(x, y)$ in the region $\mathbf{S}_{i,j}$, where $i, j = 2, \ldots, n-2$. We divide the set of representative points into three subsets $\mathcal{RP}_1 = \{rp_1, rp_2, \ldots, rp_{i-1}, rp_i\}$, $\mathcal{RP}_2 = \{rp_{i+1}, \ldots, rp_j\}$, and $\mathcal{RP}_3 = \{rp_{j+1}, rp_{j+2}, \ldots, rp_n\}$. Each point in $\mathcal{RP}_2$ has the cost value no smaller than c_{i+1} and the delay value no smaller than d_j, so, any point in the region $[c_i, c_{i+1}] \times [d_{j+1}, d_j]$ is more representative than each point in $\mathcal{RP}_2$. That means the feasible region provided by the point (x, y) must include the feasible region provided by all the points in $\mathcal{RP}_2$.

For example, in Fig. 8, suppose that we compute the local optimal point in the region $\mathbf{S}_{2,3} = [c_2, c_3] \times [d_4, d_3]$, we thus have $\mathcal{RP}_1 = \{(c_1, d_1), (c_2, d_2)\}$, $\mathcal{RP}_2 = \{(c_3, d_3)\}$, and $\mathcal{RP}_3 = \{(c_4, d_4), (c_5, d_5)\}$. The feasible region of the point (c_3, d_3) is included by the feasible region provided by any point in $\mathbf{S}_{2,3}$.

We compute the feasible region provided by $\mathcal{RP}_1$ while not included by the point (x, y) as follows:

$$\Delta_+^1(rp) = \sum_{k=2}^{i} (d_{k-1} - d_k)(x - c_k). \tag{11}$$

The feasible region provided by $\mathcal{RP}_3$ while not included by the point (x, y) is

$$\Delta_+^2(rp) = \sum_{k=j+1}^{n-1} (c_{k+1} - c_k)(y - d_k). \tag{12}$$

For example, in Fig. 8, we compute $\Delta_+^1(ap) = (x - c_2)(d_1 - d_2)$ and $\Delta_+^2(ap) = (c_5 - c_4)(y - d_4)$. We thus obtain $\Delta_+(ap) = \Delta_+^1(ap) + \Delta_+^2(ap)$.

Now, we compute the infeasible region induced by ap. The approximated feasible region defined by ap is $\mathbf{R}_x = [x, c_n] \times [y, d_1]$. Denote $\mathbf{R}_e$ as the real feasible region included in $\mathbf{R}_x$. $\mathbf{R}_x \setminus \mathbf{R}_e$ is thus the infeasible region induced by ap.

Define a point $p_c = (x, d_i)$ and $p_d = (c_{j+1}, y)$, which are on the staircase, as illustrated in Fig. 8. The feasible region $\mathbf{R}_e$ is thus defined by p_c, the points in $\mathcal{RP}_2$, and p_d. For example, in Fig. 8, $\mathbf{R}_e$ is defined by the points $\{p_c, (c_3, d_3), p_d\}$. We thus compute the infeasible region induced by the point ap as follows:

$$\begin{aligned}\Delta_-(rp) = (c_{i+1} - x)(d_i - y) + \\ \textstyle\sum_{k=i+1}^{j} (c_{k+1} - c_k)(d_k - y)\end{aligned} \tag{13}$$

For example, in Fig. 8, we compute $\Delta_-(rp) = (c_3 - x)(d_2 - y) + (c_4 - c_3)(d_3 - y)$.

We thus obtain the total approximation error as

$$f_o(x, y) = \Delta_+((x, y)) + \Delta_-((x, y)). \tag{14}$$

In addition, we have

$$\begin{cases} \frac{\partial f_o(x,y)}{\partial x} = d_1 - 2d_i + y. \\ \frac{\partial f_o(x,y)}{\partial y} = c_n - 2c_{j+1} + x. \end{cases} \tag{15}$$

We now compute the Hessian matrix of $f_o(x, y)$ as follows.

$$\mathbf{H}(f_o) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{16}$$

Since the Hessian matrix of $f_o(x, y)$ is positive definite, we can use the Kuhn-Tucker method, as referred to page 670-677 in [17], to compute the optimal values of c_x and d_x. We must consider the following cases:

1) $\frac{\partial f_o(c_x,d_x)}{\partial c_x} = 0$ and $\frac{\partial f_o(c_x,d_x)}{\partial d_x} = 0$
2) $c_x = c_i$ or c_{i+1} and $d_x = d_j$ or d_{j+1}
3) $f'_o(c_i, d_x) = 0$
4) $f'_o(c_{i+1}, d_x) = 0$
5) $f'_o(c_x, d_j) = 0$
6) $f'_o(c_x, d_{j+1}) = 0$

By (15), the process of computing the local optimal point in the region $[c_i, c_{i+1}] \times [d_{j+1}, d_j]$ is as follows. If $2rp_{j+1}.c - rp_n.c$ falls between c_i and c_{i+1}, we let $x = 2rp_{j+1}.c - rp_n.c$, otherwise, we let x be c_i or c_{i+1}. If $2rp_i.d - rp_1.d$ falls between d_{j+1} and d_j, we let $y = 2rp_i.d - rp_1.d$, otherwise, we let y be d_j or d_{j+1}. There are maximally four possible local optimal points in the region $\mathbf{S}_{i,j}$, we thus select the one with the minimum objective function as the local optimal points in $\mathbf{S}_{i,j}$.

B. Multiple points approximation

In this section, we consider the case that $\mathcal{H} > 3$. Assume that ap_m and ap_{m+1}, where $m = 1, \ldots, \mathcal{H}-1$, are located in the regions $[c_i, c_{i+1}] \times [d_{j+1}, d_j]$ and $[c_k, c_{k+1}] \times [d_{l+1}, d_l]$, respectively. By Lemma 3, we have $i \leq j$ and $l \leq k$. We will show that $k \geq j + 1$ as follows.

Lemma 4: Given any two successive representative points ap and ap' in the set of the approximated points $\mathcal{AP}$, where $ap.c < ap'.c$, assume that ap and ap' are located in the regions $[c_i, c_{i+1}] \times [d_{j+1}, d_j]$ and $[c_k, c_{k+1}] \times [d_{l+1}, d_l]$, respectively. We have $k \geq j + 1$.

Proof: We prove by contradiction. Assume that $k < j + 1$. Define a point $p_0 = (c_{j+1}, ap'.d)$. Define R_1 as the supported QoS defined by ap and ap', and R_2 as the supported QoS defined by ap and p_0. We thus have $R_1 \setminus R_2 = [ap'.c, c_{j+1}] \times [ap'.d, ap.d]$. Since $ap.d < d_j$, $R_1 \setminus R_2$ is actually the infeasible region. Therefore, the approximation error produced by $\{ap, ap'\}$ is greater than $\{ap, p_0\}$.

The lemma is thus proved. ■

For example, in Fig. 9, suppose that $ap = ap_1$ and $ap' = ap_2$. p_1 and p_2 are located in the regions $[c_2, c_3] \times [d_5, d_4]$ and $[c_3, c_4] \times [d_6, d_5]$, respectively. In this case, we can find a point $p_0 = (c_5, p_2.d)$. Denote R_1 as the approximated feasible region defined by p_1 and p_2, and R_2 as the approximated feasible region defined by p_1 and p_0. We thus compute that $R_1 \setminus R_2 = [p_2.c, c_5] \times [p_0.d, p_1.d]$, which is actually

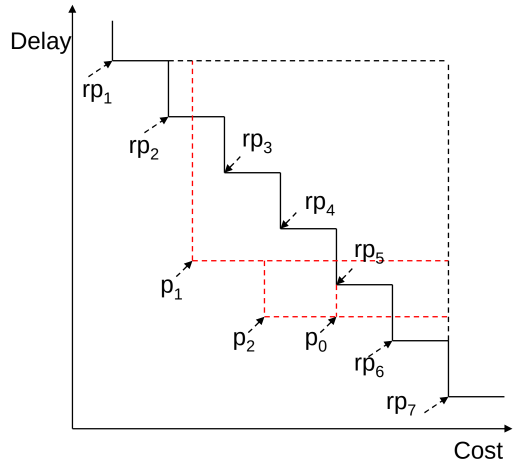


Fig. 9. Illustration for Lemma 4.

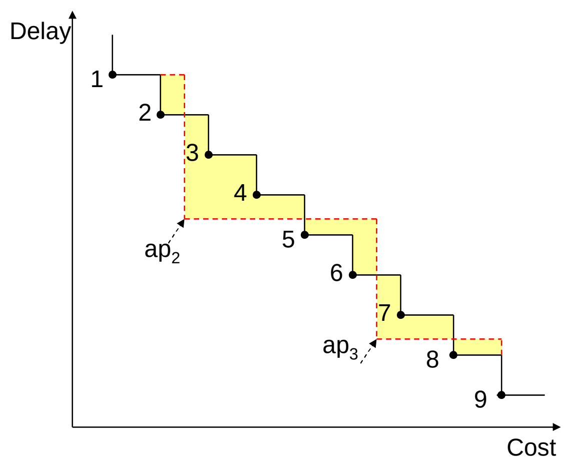


Fig. 10. An illustration for staircase approximation with multiple points.

the infeasible region. That means the approximation error produced by p_1 and p_2 are greater than that by p_1 and p_0. Therefore, p_1 and p_2 cannot be contained simultaneously in $\mathcal{AP}$.

As we mentioned in the previous subsection, it is intractable to give a general formulation representing the approximation error produced by $\mathcal{AP}$. By Lemma 4, each $\mathbf{S}_{i,j}$, where $i, j = 2, \ldots, \mathcal{H}-1$, at most contain one approximated points in $\mathcal{AP}$. We randomly select $\mathcal{H}-2$ subregions in $\mathbf{S}$ and assume that each subregion contains one approximated point. We then compute a local optimal $\mathcal{AP}$ under this specific combination. After trying all the possible combinations, we will obtain the global optimal $\mathcal{AP}$. Let N be the number of the subregions, the number of combinations is less than $\binom{N}{\mathcal{H}-2}$. Therefore, we need to compute at most $\binom{N}{\mathcal{H}-2}$ local optimal $\mathcal{AP}$ in order to get the global optimal approximated staircase. We are going to discuss how to compute a local optimal $\mathcal{AP}$ with a given combination.

Note that $ap_1 = rp_1$ and $ap_{\mathcal{H}} = rp_n$, we can say that ap_1 is located in the region $\mathbf{S}_{1,1} = [c_1, c_2] \times [d_2, d_1]$ and $ap_{\mathcal{H}}$ is located in the region $\mathbf{S}_{n-1,n-1} = [c_{n-1}, c_n] \times [d_n, d_{n-1}]$. Given $\mathcal{H}-2$ subregions selected from

$\{\mathbf{S}_{i,j}, i, j = 2, \ldots, n-2\}$, we assume that ap_m is located in the region $[c_{i_m}, c_{i_m+1}] \times [d_{j_m+1}, d_{j_m}]$, where $i_m, j_m = 2, \ldots, n-2$ and $m = 1, \ldots, \mathcal{H}$. We have $j_m \leq i_m$ and $i_{m+1} > j_m$ for all $m = 2, \ldots, \mathcal{H}-1$. We also have $i_1 = j_1 = 1$ and $i_{\mathcal{H}} = j_{\mathcal{H}} = n-1$.

For example, in Fig. 10, $\mathcal{RP}$ contains nine points and $\mathcal{AP}$ contains four points. We select the subregions $\mathbf{S}_{2,4} = [c_2, c_3] \times [d_5, d_4]$ and $\mathbf{S}_{6,7} = [c_6, c_7] \times [d_8, d_7]$. ap_2 are thus located in $\mathbf{S}_{2,4}$ and $\mathbf{S}_{6,7}$, respectively. Based on this specific combination, we have $i_2 = 2$, $j_2 = 4$, $i_3 = 6$, and $j_3 = 7$. Given a specific combination, we are going to discuss how to compute the approximation error produced by $\mathcal{AP}$.

According to the approximated points set $\mathcal{AP}$, we divide the region $[c_1, c_n] \times [d_1, d_n]$ into a set subregions $\{\mathbf{A}_m = [c_1, c_n] \times [y_m, y_{m-1}]\}$, where $m = 2, \ldots, \mathcal{H}$. We then compute the approximation error in each region $\mathbf{A}_m$.

The real representative points located in the region $\mathbf{A}_m$ are $\mathcal{RP}_m = \{(rp_{j_{m-1}+1}, \ldots, rp_{j_m})\}$. For example, in Fig. 10, let $m = 2$, and we compute the approximation error in the region $[c_1, c_9] \times [ap_2.d, d_1]$, where $ap_1.d = d_1$. Since $j_{m-1} = 1$ and $j_m = 4$, the set of representative points located in the region $[c_1, c_9] \times [d_1, ap_2.d]$ is $\{rp_2, rp_3, rp_4\}$. Let $m = 3$, the set of representative points in the region $[c_1, c_9] \times [ap_3.d, ap_2.d]$ is $\{rp_5, rp_6, rp_7\}$ since $j_2 = 4$ and $j_3 = 7$. Let $m = 4$, the set of representative point in the region $[c_1, c_9] \times [d_4, ap_3.d]$, where $d_4 = ap_4.d$, is $\{rp_8\}$.

Similarly as in the previous section, we divide $\mathcal{RP}_m$ into two subsets $\mathcal{RP}_{m,1} = \{rp_{j_{m-1}+1}, \ldots, rp_{i_m}\}$ and $\mathcal{RP}_{m,2} = \{rp_{i_m+1}, \ldots, rp_{j_m}\}$. Each point in $\mathcal{RP}_{m,2}$ has the cost value no smaller than c_{i_m+1} and the delay value no smaller than d_{j_m}, so, the point (x_m, y_m) must be more representative than each point in $\mathcal{RP}_{m,2}$. In Fig. 10, $\mathcal{RP}_{2,1} = \{rp_2\}$ and $\mathcal{RP}_{2,2} = \{rp_3, rp_4\}$, since $i_2 = 2$. Since $i_3 = 6$, we have $\mathcal{RP}_{3,1} = \{rp_5, rp_6\}$ and $\mathcal{RP}_{3,2} = \{rp_7\}$. Since $i_{\mathcal{H}} = j_{\mathcal{H}}$, it holds that $\mathcal{RP}_{\mathcal{H},2} = \varnothing$. For example, in Fig. 10, $\mathcal{RP}_{4,1} = \{rp_8\}$ and $\mathcal{RP}_{4,2} = \varnothing$.

The overestimated areas in the region $\mathbf{A}_m$, denoted by Δ_+^m, is the difference of the feasible region provided by the representative points in $\mathcal{RP}_{m,1}$ and that by the point (x_m, y_m). With the similarity to (11), Δ_+^m can be computed as follows:

$$\begin{aligned}\Delta_+^m = &(y_{m-1} - d_{j_{m-1}+1})(x_m - c_{j_{m-1}+1}) + \\ &\textstyle\sum_{t=j_{m-1}+2}^{i_m} (d_{t-1} - d_t)(x_m - c_t).\end{aligned} \tag{17}$$

The underestimated areas in the region $\mathbf{A}_m$, denoted by Δ_-^m, is the difference of the feasible region provided by (x_m, y_m) and that by the points in $\mathcal{RP}_{m,2}$. With the similarity to (12), Δ_-^m can be computed as follows:

$$\Delta_{-}^{m} = (c_{i_m+1} - x_m)(d_{i_m} - y_m) + \sum_{t=i_m+1}^{j_m} (c_{t+1} - c_t)(d_t - y_m) \tag{18}$$

Since $\mathcal{RP}_{\mathcal{H},2} = \varnothing$, it holds that $\Delta_{-}^{\mathcal{H}} = 0$.

For example, in Fig. 10, let $m = 2$. We consider the approximation error in the region $[c_1, c_9] \times [y_2, y_1]$, where $y_1 = d_1$ and $y_2 = ap_2.d$. Since $i_m = 2$ and $j_{m-1} = 1$, by (17), we obtain $\Delta_{+}^{2} = (d_1 - d_2)(x_2 - c_2)$. By (18), we obtain $\Delta_{-}^{2} = (c_3 - x_2)(d_2 - y_2) + (c_4 - c_3)(d_3 - y_2) + (c_5 - c_4)(d_4 - y_2)$. With the same method, we obtain that $\Delta_{+}^{3} = (y_2 - d_5)(x_3 - c_5) + (d_5 - d_6)(x_3 - c_6)$ and $\Delta_{-}^{3} = (c_7 - x_3)(d_6 - y_3) + (c_8 - c_7)(d_7 - y_3)$. We can also obtain that $\Delta_{+}^{4} = (y_3 - d_8)(c_9 - c_8)$ and $\Delta_{-}^{4} = 0$. The shaded areas in Fig. 10 are the total approximation error produced by $\mathcal{AP}$.

We thus obtain the nonlinear programming problem stated as follows:

$$\begin{aligned} &\min f_o = \sum_{m=2}^{\mathcal{H}} \Delta_{+}^{m} + \Delta_{-}^{m} \\ &\text{s.t.} \quad c_{i_m} \le x_m \le c_{i_m+1} \\ &\qquad\; d_{j_m+1} \le y_m \le d_{j_m} \\ &\qquad\; m = 2, 3, \ldots, \mathcal{H} - 1 \end{aligned} \tag{19}$$

f_o is the function of the variables $\{(x_m, y_m), m = 2, \ldots, \mathcal{H} - 1\}$. For all $m = 2, \ldots, \mathcal{H} - 1$, we have

$$\begin{cases} \frac{\partial f_o}{\partial x_m} = y_{m-1} + y_m - 2d_{i_m}. \\ \frac{\partial f_o}{\partial y_m} = x_{m+1} + x_m - 2c_{j_m+1}. \end{cases} \tag{20}$$

We now compute the Hessian matrix of $f_o(x, y)$ as follows.

$$\mathbf{H}(f_o) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \ldots & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 1 & 0 \end{pmatrix}. \tag{21}$$

$\mathbf{H}(f_o)$ is an $(\mathcal{H}-2)\times(\mathcal{H}-2)$ matrix. We can easily verify that all the $\mathcal{H}-2$ eigenvalues are 1. Since the Hessian matrix of f_o is positive definite, so that we can use Kuhn-Tucker method to find the optimal values of $\{(x_m, y_m), m = 2, \ldots, \mathcal{H}-1\}$. We discuss the process of computing these optimal values as follows.

First, we let $\frac{\partial f_o}{\partial x_m} = 0$ and $\frac{\partial f_o}{\partial y_m} = 0$ for all $m = 2, \ldots, \mathcal{H}-1$. We thus get

$$\begin{cases} x_m = 2c_{j_m+1} - x_{m+1} \\ y_m = 2d_{i_m} - y_{m-1} \end{cases} \tag{22}$$

Since $x_{\mathcal{H}} = c_n$ and $y_1 = d_1$, we can easily compute the values of x_m and y_m for all $m = 2, \ldots, \mathcal{H}-1$ based on (22).

If $c_{i_m} \leq x_m \leq c_{i_m+1}$ and $d_{j_m+1} \leq y_m \leq d_{j_m}$ for all $m = 2, \ldots, \mathcal{H}-1$, we have obtained the optimal values of $\mathcal{AP}$.

If $\exists m$ such that x_m is not located in the range $[c_{i_m}, c_{i_m+1}]$, we set x_m be c_{i_m} or c_{i_m+1}. We then compute the values of x_k for all $k = m-1, m-2, \ldots, 2$ based on the value of x_m. This process continues until all the values of $\{x_m, m = 2, \ldots, \mathcal{H}-1\}$ satisfy the corresponding constraints.

With the same method, if $\exists m$ such that y_m is not located in the range $[d_{j_m+1}, d_{j_m}]$, we set y_m be d_{j_m} or d_{j_m+1}. We then compute the values of y_k for all $k = m+1, \ldots, \mathcal{H}-1$. This process continues until all the values of $\{y_m, m = 2, \ldots, \mathcal{H}-1\}$ satisfy the corresponding constraints. Therefore, we will maximally obtain $2^{\mathcal{H}-1}$ possible optimal points, then, we select the one with the minimum objective function.

C. Discussion

Assume that border bs has $\mathcal{A}(bs)$ border neighbors. Each neighbor j of bs advertises the supported QoS from itself to a destination, which is approximated by at most $\mathcal{H}$ representative points. Moreover, bs has computed the supported QoS from itself to j, which is also approximated by at most $\mathcal{H}$ representative points. Therefore, the supported QoS from bs to a destination, via j is approximated at most nine representative points. The total supported QoS from bs to a destination is thus approximated at most $\mathcal{O}(\mathcal{A}(bs))$ representative points. Denote n as the number of the representative points defining the supported QoS. After obtaining the supported QoS to a destination, bs will find $\mathcal{H}$ approximated representative points to approximate the supported QoS.

Let (c_l, d_u) and (c_u, d_l) be the minimum cost and the minimum delay QoS parameters, respectively. As mentioned before, in order for computing the optimal $\mathcal{H}-2$ approximated representative points, we first divide the infeasible region included in $[c_l, c_u] \times [d_l, d_u]$, into $N = \frac{(n-2)(n-3)}{2}$ subregions. We mentioned earlier that each subregion at most contains one approximated point. We select $\mathcal{H}-2$ subregions and assume that each approximated point is located in each of them. The maximal number of $(\mathcal{H}-2)$-combination is less than $\binom{N}{\mathcal{H}-2}$. That means we will get maximally $\binom{N}{\mathcal{H}-2}$ local optimal approximated staircases, and then find the global optimal $\mathcal{AP}$ with the minimum approximation error. Therefore, the time complexity of border node bs computing the supported QoS is $\mathcal{O}(|\mathcal{A}(bs)|^{2(\mathcal{H}-2)})$.

Since the aggregation method may underestimate the minimum delay for some cost values, so that the algorithm induces some infeasible region. Therefore, it is possible that the our routing protocol may accept some infeasible requests, but cannot find a feasible physical path for them. In our routing protocol, the request keeps track the accumulated QoS metrics as it transmitting across the network. When an intermediate node detects that the accumulated delay or cost exceeds the requirements, it will drop this request. We call such request *crankbacked request*.

Moreover, the underestimation situation may cause the occurrence of the route oscillation, as referred to [2], [14]. The route oscillation makes our routing algorithm diverge, we thus apply the *advertisement history checking* method [14] to suppress the advertisement, which is discussed as follows.

For a newly computed QoS, if several similar QoSes (say, two) have been advertised, this newly computed QoS will not be advertised. The comparing of the two QoSes is as follows:

All the supported QoS contains the minimum cost representative point and the minimum delay representative point. If the newly computed minimum cost or the minimum delay representative point is different than that of the old one, these two QoSes are different. Otherwise, we need to compare the newly computed aggregated representative point rp_{new} to the old one rp_{old}. Given a threshold t, the allowed variation of the cost metric is defined as $v_c = \min(rp_{new}.c, rp_{old}.c) \cdot t$, and that of the delay metric is defined as $v_d = \min(rp_{new}.d, rp_{old}.d) \cdot t$. If the difference of the cost metrics of rp_{new} and rp_{old} is less than v_c and the difference of the delay metrics is less than v_d, these two QoSes are similar.

Due to the route oscillation, there may exist loops in the routing tables of some border nodes. Therefore, each request also keeps track the path information it traversed, if an intermediate node detects that there exists loop, it will drop this request. We also call such request crankbacked request.

VI. SIMULATION

In this section, we present our simulation results and compare the proposed topology QoS information aggregation method, called SCAM, to the existing line segment method, called LSAM. It is obvious that as the number of the approximated representative points increases, the approximation error is smaller. However, the space requirement and the computational overhead increases. In our simulation, the supported QoS is approximated with three points.

A. Simulation testbed

We evaluate the performance of the algorithms by using a self-written C++ network simulator. The network topology is generated by BRITE [11]. The intradomain topology is generated based on the Waxman's model [11], and the interdomain topology is generated based on the Barabasi-Albert model [11]. All physical link metrics are asymmetric. Each link (i, j) is associated with two randomly generated metrics, delay and cost. These metrics are uniformly distributed, such that $c_{i,j} \sim \mathrm{uniform}[1, 100]$ and $d_{i,j} \sim \mathrm{uniform}[1, 300]$.

In the networks, each domain contains 50 nodes, the average border number in each domain is four, and the average number of interdomain links is about four times the domain number. We simulate asynchronous advertisement exchange in interdomain routing and each advertisement transmission delay is a random time unit selected in the range $(0.0, 2.0]$. Each advertisement message contains the supported QoS information from a border node to a destination domain. The threshold value, t, for comparing two different QoS parameter pairs is set to be 5%. Given the lower bound and upper bound of the cost constraint C_l and C_u, and the lower bound and the upper bound of the delay constraint D_l and D_u, we call the region $[C_l, C_u] \times [D_l, D_u]$ as the QoS request space. We have generated ten requests for each source-border-to-destination-domain pair with the cost requirement $c_{req} \sim \mathrm{uniform}[C_l, C_u]$ and the delay requirement $d_{req} \sim \mathrm{uniform}[D_l, D_u]$. We would like to investigate the behavior of the interdomain routing protocols by changing the QoS request space.

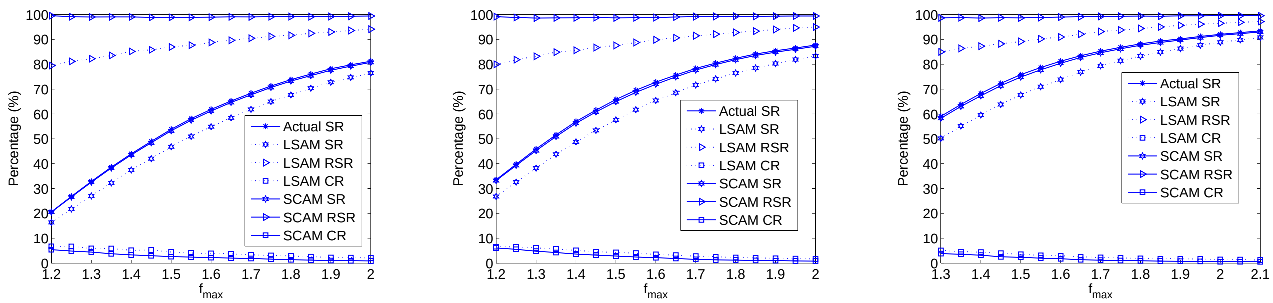
B. Evaluation metrics

Four performance metrics are used in our evaluation: *success ratio (SR)*, *crankback ratio (CR)*, *convergence speed*, and *advertisement overhead*.

Success ratio: There are two kinds of success ratio: absolute success ratio and relative success ratio. The absolute success ratio refers to the ratio of the connection requests for which feasible paths are found to the total number of the receiving connection requests. The relative success ratio is the ratio of the absolute success ratio produced by the routing protocol with topology aggregation to that without topology aggregation. Therefore, the smaller Δ_+, the greater the relative success ratio.

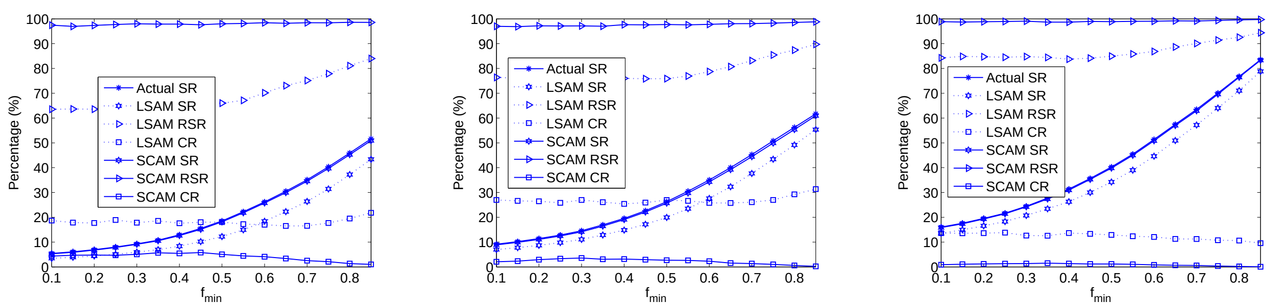
Crankback ratio is defined as the ratio of the number of the crankbacked requests to the number of the accepted requests. We mentioned that there are two kinds of crankbacked request: the infeasible request accepted by the source node while dropped by the intermediate node and the feasible request for which the routing protocol cannot establish a feasible physical path.

Convergence speed refers to the time period from the first advertisement message to the last advertisement message in the network. *advertisement overhead* is represented by the total number of the messages generated in the network.



(a) 10-domain networks with $f_{\min} = 1.0$. (b) 10-domain networks with $f_{\min} = 1.1$. (c) 10-domain networks with $f_{\min} = 1.2$.

Fig. 11. The performance of the interdomain algorithms by changing $f_{\max}$.



(a) 10-domain networks with $f_{\max} = 0.9$. (b) 10-domain networks with $f_{\max} = 1.0$. (c) 10-domain networks with $f_{\max} = 1.1$.

Fig. 12. The performance of the interdomain algorithms by changing $f_{\min}$.

C. Simulation results

In this section, LSAM SR and SCAM SR are the approximated absolute success ratio produced by the algorithm LSAM and SCAM, respectively. LSAM RSR and SCAM RSR are the relative success ratio delivered by LSAM and SCAM, respectively. LSAM CR and SCAM CR are the crankbacke ratio produced by LSAM and SCAM.

Given a source and a destination domain, define (c_l, d_u) and (c_u, d_l) as the QoS parameter pair of the minimum cost path and the minimum delay path, respectively. We first set the QoS request space to be

$[f_{\min}c_l, f_{\max}c_l] \times [f_{\min}d_l, f_{\max}d_l]$. We fix $f_{\min}$ and change $f_{\max}$. The simulation results are presented in Fig. 11. We then set the QoS request space to be $[f_{\min}c_u, f_{\max}c_u] \times [f_{\min}d_u, f_{\max}d_u]$. We fix $f_{\max}$ and change $f_{\min}$. The simulation results are shown in Fig. 12. From these simulation results, we can see that the relative success ratio produced by our algorithm is higher than that by LSAM and the crankback ratio delivered by our algorithm is lower than that by LSAM. In addition, we observe that the relative success ratio produced by our algorithm is near to 1 and the crankback ratio is less than 7%, whatever the actual absolute success ratio is. However, the performance of LSAM changes obviously as the QoS request space changes. For instance, in Fig. 12(a), when $f_{\max} = 0.9$ and $f_{\min} \leq 0.4$, the success ratio of LSAM is less than 70% and the crankback ratio is about 20%. In Fig. 12(b), when $f_{\max} = 1$, the crankback ratio of LSAM is about 30%. Therefore, the performance of our algorithm is more stable than that of LSAM with the changes of the QoS request space.

We mentioned that the optimization objective of the QoS information aggregation method is to minimize the overestimation area Δ_+ and the underestimation area Δ_-. We compute the area of the feasible region included in $[c_l, c_u] \times [d_l, d_u]$, denoted by Δ. $\frac{\Delta_+}{\Delta}$ is defined as the overestimation area ratio (OR) and $\frac{\Delta_-}{\Delta}$ is defined as the underestimation area ratio (UR). $\frac{\Delta_+ + \Delta_-}{\Delta}$ is defined as the approximation error (AE) produced by the aggregation algorithms. Fig. 13 presents the approximation error produced by LSAM and SCAM under different network topologies. The simulation results show that the overestimation area ratio produced by our approach is much smaller than that by LSAM and the underestimation area ratio is a little higher than that by LSAM. In conclusion, the approximation error produced by our approach is much smaller than that by LSAM. We also observe that the approximation error produced by our approach and LSAM changes a little as the network size increases, which demonstrates that the proposed aggregation method is suited for the large-scale networks.

Table I shows the convergence speed of LSAM and SCAM, and Table II shows the the number of the messages generated in the computation process. We observe that our approach has faster convergence speed than SCAM, and the the number of the message generated by our approach is smaller than that by SCAM. Although each message in our approach contains six numbers while that in LSAM contains four numbers, we still can say that our approach has smaller advertisement overhead than LSAM, since it is well known that advertising more messages consumes much more resources than increasing the message size.

Domain Number	LSAM	SCAM
10	45.474	36.92
20	110.674	83.6538
30	213.993	140.87
40	352.316	216.567
50	477.139	288.428

TABLE I
CONVERGENCE SPEED (TIME UNITS).

Domain Number	LSAM	SCAM
10	2284.5	1963.3
20	15065.6	11883.9
30	45984	33727.8
40	99471.6	70034.4
50	175041	120219

TABLE II
ADVERTISEMENT OVERHEAD.

VII. CONCLUSION

In this paper, we investigate a very challenging problem, supporting QoS routing in the Internet. The challenge, scalability, makes this problem very difficult to solve. Accordingly, we propose a novel QoS information aggregation method. In our approach, a set of approximated QoS parameters, which has constant size, is used to approximate the supported QoS, so that the advertisement message size and the storage requirement is independent of the network size and topology. By extensive simulation, we show that our approach can serve the requests more efficiently than the existing method. The approximation error produced by our approach is much smaller than that of the existing method. Moreover, the network size

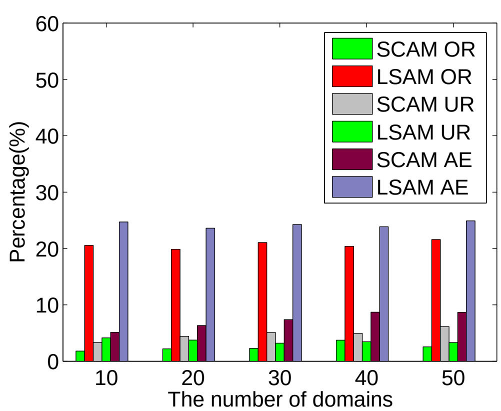


Fig. 13. The performance of the approximation error.

and the QoS request space almost cannot affect the performance of our algorithm, which demonstrates that our algorithm is scalable and stable. More generally, the proposed aggregation method can be extended for aggregating the QoS information with additive-concave or concave-concave constraints.

REFERENCES

[1] F. Hao and E. Zegura, "On scalable QoS routing: performance evaluation of topology aggregation," *INFOCOM'97*, pp. 147-156, 2000.

[2] R. Hou, K.-S. Lui, K.-C. Leung, and F. Baker,"Quality-of-Service Routing with Two Additive Constraints in Hierarchical Networks," *submitted for publication*.

[3] A. Iwata, H. Suzuki, R. Izmailow, and B. Sengupta, "QoS aggregation algorithms in hierarchical ATM networks," *IEEE Proceedings of the International Conference (ICC'98)*, pp. 243-248, 1998.

[4] S. Jeon, "Topology aggregation method for multiple link parameters," *Globecom*, pp. 1378-1382, 2004.

[5] T. Korkmaz and M. Krunz, "Source-oriented topology aggregation with multiple QoS parameters in hierarchical netowrks," *ACM Transactions on Modeling and Computer Simulation*, vol. 10, no. 4, pp. 295-325, October 2000.

[6] K.-C. Leung, K.-S. Lui, K.-C. Leung, and F. Baker, "Quality-of-Service Routing with Two Concave Constraints," *2008 IEEE International Conference on Communications (ICC 2008)*, Beijing, China, May 2008.

[7] K.-C. Leung, Ronghui Hou, K.-S. Lui, and K.-C. Leung, "Quality-of-Service Routing with Two Concave Constraints in Internet," *submmitted for publishing*.

[8] K.-S. Lui, K. Nahrstedt, and S. Chen, "Routing with topology aggregation in delay-bandwidth sensitive networks," *IEEE/ACM Transactions on Networking*, 12(1): 17-29, February 2004.

[9] W. Lee, "Spanning tree method for link state aggregation in large communication networks," in *Proc. IEEE INFOCOM*, pp. 297-302, 1995.

[10] W. Lee, "Topology aggregation for hierarchical routing in ATM networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 25, pp. 82-92, April 1995.

[11] A. Medina, A. Lakhina, I. Matta, J. Byers, "Brite: an approach to universal topology generation," *Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems - MASCOTS'01*, pp. 346-353, August 2001.

[12] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," *Network Working Group*, RFC 1771, Mar. 1995.

[13] V. Sarangan, D. Ghosh, and R. Acharya, "Capacity-aware state aggregation for interdomain QoS routing," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 792-808, August 2006.

[14] W.-Y. Tam, K.-S. Lui, S. Uludag, K. Nahrstedt, "Quality-of-service routing with path information aggregation," *Computer Networks*, 51(12): 3574-3594, August 2007.

[15] Y. Tang, S. chen, "QoS information approximation for aggregated networks," *IEEE International Conference on Communication, ICC'04*, vol. 4, pp. 2107-2111, June 2004.

[16] S. Uludag, K.-S. Lui, K. Nahrstedt, G. Brewster, "Analysis of Topology Aggregation Techniques for QoS Routing," *ACM Computing Surveys (CSUR)*, vol. 39, no. 3, Article No. 7, 2007.

[17] W. L. Winston, *Operations Research: Applications and Algorithms.* Duxbury Press; 4 edition, July, 2003.

[18] L. Xiao, J. Wang, K.-S. Lui, K. Nahrstedt, "Advertising Inter-domain QoS Routing Information," *IEEE Journal on Selected Areas in Communications*, 22(10): 1949-1964, December 2004.

[19] T. Zhang, Y. Cui, Y. Zhao, L. Fu, T. Korkmaz, “Scalable BGP QoS Extension with Multiple Metrics,” *2006 International Conference on Networking and Services (ICNS’06)*, pp. 80-85, 2006.