

Routing with QoS Information Aggregation in Hierarchical Networks

Ronghui Hou, King-Shan Lui, Ka-Cheong Leung
Department of Electrical and Electronic Engineering
The University of Hong Kong
Hong Kong SAR, China
E-mail: {rhhou, kslui, kcleung}@eee.hku.hk

Fred Baker
Cisco Research Center
170 West Tasman Dr.
San Jose, CA 95134, USA
E-mail: fred@cisco.com

Abstract—In this paper, we consider the problem of routing with two additive constraints in the hierarchical networks, such as the Internet. In order for scalability, the supported QoS information in the hierarchical networks has to be aggregated. We propose a novel method for aggregating the QoS information. 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. We use a set of real numbers to approximate the supported QoS between different domains. The size of the set is predefined so that advertisement overhead and the space requirement will not grow exponentially as the network size grows. The simulation results show that the proposed method outperforms the existing methods.¹

Index Terms—QoS routing, hierarchical networks, additive constraints, QoS information 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 state information of other domains. For computing the supported QoS between two different domains, each border node has to advertise the supported QoS information from itself to a destination to its border neighbors, so that the border neighbors can obtain the supported QoS to the same destination. We use the following example to illustrate this problem.

Fig. 1 shows a simple network 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 node $X.i$ means the border node i in domain X . Now, we consider the process of computing the supported QoS from $S.2$ to the domain T . $B.2$ and $C.2$ are directly connected to domain T and they advertise the supported QoS from itself to T 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 .

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 , respectively. For a path $p = \{l_1, l_2, \dots, l_h\}$, 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. A path with the QoS parameter (c_p, d_p) can satisfy the QoS requirement of a request (c, d) , where $c \geq c_p$ and $d \geq d_p$. We call a request that can be supported by a path a *feasible request*. Geometrically, any feasible request supported by (c_p, d_p) has the QoS requirement falling into the upper right quadrant of the point. If there are several paths going from one node to the other, the feasible requests supported by these paths form a staircase. Fig 2 shows an example to the *feasible region* when there are four paths and the QoS parameter of these paths are p_1, p_3, p_5 , and p_7 . However, it is not scalable to advertise a staircase, which is defined by a set of points. To solve the problem, an approximation of the supported QoS region is advertised instead [15], [17]. This process of approximation is called *QoS information aggregation*.

There have been a few works on the QoS information aggregation. Generally, the works in [1], [3], [9], [13] just consider one QoS metric. The logical link between any two border nodes is selected as the “best” path by using the shortest-path algorithm. The work in [19] studies the advertisement of the QoS information between two different domains under the condition of the inaccurate link state. Instead of using a single value, it employs a statistical distribution to represent a single metric of the path between any two border nodes. 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. It also could not provide a polynomial-time routing algorithm corresponding to this method. The work in [4] considers multiple QoS parameters. the QoS parameter of each logical link between any two borders is set to be the best one, that is the minimum delay and cost metrics of all paths between both the border nodes, while in [11], the QoS parameter of each logical link between any two border nodes is set to be the worst one, that is the maximum delay and cost metrics for all the paths between two border nodes. In [7], it has been shown that the best-case point or the worst-case point does not work well. The work in [7] was the first to propose a method to aggregate the supported QoS and the corresponding

¹This work is supported in part by the Cisco Research Initiative Award.

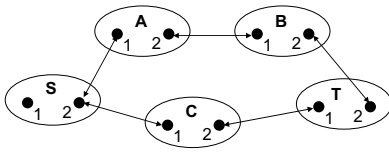


Fig. 1. A simple network.

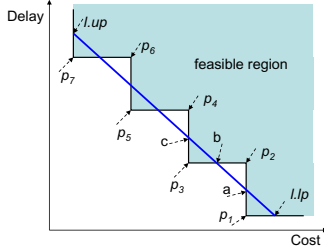


Fig. 2. The line segment approximation scheme.

polynomial-time routing algorithm. In [7], a line segment is used to approximate the staircase, as illustrated in Fig. 2. The works in [6], [14] study the distance-vector routing problem with additive-concave and concave-concave metrics based on the line segment aggregation method. However, the line segment approach used in [7] is computed by the method of least squares, which does not correspond to the actual optimization problem of the QoS information aggregation. The works in [15], [16] also proposed several approaches for aggregating the supported QoS. However, all the solutions are found by the method of least squares, moreover, they did not give the polynomial-time routing algorithm based on the proposed QoS information aggregation mechanisms.

In this paper, we propose a novel method of representing the aggregated state by \mathcal{H} points where \mathcal{H} is predefined by the system, so that it is scalable for advertising the aggregated supported QoS. To the best of our knowledge, we are the first to study the *area-minimization optimization*, the de facto optimization problem for the QoS information aggregation. We evaluate the performance of our algorithm. The simulation results show that our algorithm outperforms the existing algorithm due to smaller distortion and fewer advertisement overhead.

II. PROBLEM FORMULATION

As mentioned in the previous section, the supported QoS forms a staircase. A line segment was proposed in [7] to approximate this staircase. Since a line segment is just an approximation of the supported QoS, it may introduce some distortion. For example, in Fig. 2, Δbcp_3 is a feasible area of the staircase but is not included in the feasible region defined by the line segment l . In that area, the minimum delay at a particular cost is *overestimated* by the line segment. We thus call this region an *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 included in the feasible region

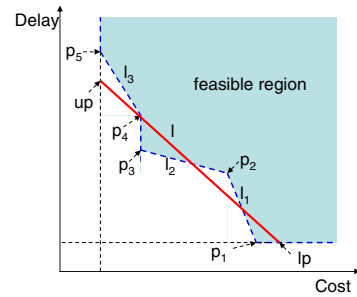


Fig. 3. An illustration for the aggregated supported QoS.

defined by the line segment, such as the triangle Δabp_2 in Fig. 2. These regions are called *underestimated areas*, 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. We call such optimization the *area-minimization optimization*. The work in [14] also mentioned finding a line segment l such that $\Delta_+(l) + \Delta_-(l)$ is minimized. However, [14] did not explain how to find such line segment. As a matter of fact, finding a line segment with the area minimization is a very difficult task, as discussed later. The work in [15] also proposed the area minimization objective. However, this work just presents several heuristic methods for aggregating the supported QoS.

Although it might not be very difficult to get a line segment to approximate a staircase for error minimization, the issue becomes more complicated when we consider the supported QoS across multiple domains. As described in [7] and [14], when a line segment is used to represent the supported QoS of a certain path from one border node to another, the aggregated supported QoS of the paths between two nodes becomes a *polyline* as illustrated in Fig. 3. Finding a line segment to approximate this polyline with the minimum error is a difficult task [21]. In this paper, we study another approach to approximate the supported QoS information. Instead of using a line segment, we employ an approximated staircase which can be defined by a predefined number of points. There are several advantages in using a staircase to approximate the supported QoS information. First, the real supported QoS is also defined by a set of points, and the current QoS routing protocol without QoS information aggregation does not need any changes. Second, in addition to propose an aggregation mechanism to approximate the optimal supported QoS, we also need to propose how to concatenate two different supported QoSes and aggregate multiple different supported QoSes [14]. If the approximated supported QoS is defined by points, it is very straight forward to add up the supported QoSes of two subpaths and aggregate several different supported QoSes.

Assume that the actual supported QoS is defined by a set of points $\mathcal{RP} = \{rp_1, rp_2, \dots, rp_n\}$ and the approximated supported QoS is defined by a set of points $\mathcal{AP} = \{ap_1, ap_2, \dots, ap_{\mathcal{H}}\}$. As it is very important to advertise the minimum cost and minimum delay of the paths, we put

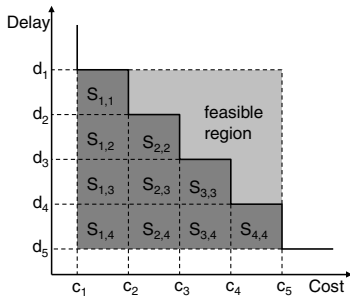


Fig. 4. An illustration for staircase approximation.

$ap_1 = rp_1$ and $ap_{\mathcal{H}} = rp_n$. That is, we only have to find the intermediate points on the approximate staircase, which are $ap_2, \dots, ap_{\mathcal{H}-1}$.

Denote the feasible region defined by \mathcal{AP} as \mathbf{R}_a and that by \mathcal{RP} as \mathbf{R}_r . $\mathbf{R}_r \setminus \mathbf{R}_a$ is the overestimated area. $\mathbf{R}_a \setminus \mathbf{R}_r$ is the underestimated area. We want to find an \mathcal{AP} such that the sum of the overestimated areas and the underestimated areas is minimized.

Suppose that $\mathcal{H} = 2$. This means that the supported QoS is just defined by the points rp_1 and rp_n . We let (c_i, d_i) be the coordinate of point rp_i on \mathcal{RP} . In this case, the actual feasible region included in $[c_1, c_n] \times [d_n, d_1]$ is not included in the region defined by \mathcal{AP} . In fact, when there are only two points in \mathcal{AP} , it never contains underestimated area, which means it does not produce any infeasible region. For example, in Fig. 4, there are five real representative points (or pareto optimal points). The light gray region and the dark gray region are the feasible region and the infeasible region included in $[c_1, c_5] \times [d_5, d_1]$, respectively.

As rp_1 and rp_n are included in \mathcal{AP} , the other points in \mathcal{AP} must have cost and delay values fall between rp_1 and rp_n . That is, the approximated points must be located in the region $[c_1, c_n] \times [d_1, d_n]$, which is the gray (light or dark) area in Fig. 4.

III. THE METHOD OF APPROXIMATE STAIRCASE

We now describe how to find an approximated staircase with the minimum approximation error in details. To facilitate our discussion, we first define the concept of *representativeness* as 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 defined by p_2 is included in that by p_1 .

Lemma 1: The optimal approximated points must 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} . Thus, we can find a representative point rp in \mathcal{RP} which is more representative than ap . rp does not induce infeasible region, but defines larger feasible region than ap . Therefore, If we replace ap with rp , the approximation error produced by \mathcal{AP} will be reduced. ■

For example, The dark gray region in Fig. 4 is the real 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 ease of discussion, we first consider the case of $\mathcal{H} = 3$. As rp_1 and rp_n are already in \mathcal{AP} , where $rp_1 = ap_1$ and $rp_n = ap_3$, we only need to find an appropriate ap_2 .

A. Three points approximation

We let ap_2 be (x, y) . We further denote the *overestimated area* introduced by \mathcal{AP} as $\Delta_+(x, y)$, and the *underestimated area* introduced by \mathcal{AP} as $\Delta_-(x, y)$. We want to find the optimal values of x and y such that $\Delta_+(x, y) + \Delta_-(x, y)$ is minimized. Since the approximation error depends on which level of the staircase ap_2 falls in, it is not easy to develop a general formula for it. We thus use the idea of *branch-and-bound*. That is, we divide the infeasible region in $[c_1, c_n] \times [d_n, d_1]$ into several rectangular regions. Then, we compute the *local* optimal point in each region. The *global* optimal point is the best point among the local optimal points.

Let the interval sets $\mathbb{S}_c = \{[c_1, c_2], \dots, [c_{n-1}, c_n]\}$ and $\mathbb{S}_d = \{[d_n, d_{n-1}], \dots, [d_2, d_1]\}$. 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 $i, j = 1, 2, \dots, n-1$. Note that, by Lemma 1, we do not need to consider the regions that are located in the actual feasible region. Therefore, we only have to consider $\mathbf{S}_{i,j}$, where $i \leq j$. For example, in Fig. 4, we only have to 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 approximated representative point must not be located in the regions $\mathbf{S}_{1,j}$ and $\mathbf{S}_{i,n-1}$ for all $i, j = 1, 2, \dots, n-1$.

Proof: Due to space limitation, we only prove the lemma for regions $\mathbf{S}_{1,j}$, where $j = 1, 2, \dots, n-1$. Given any point $p_1 = (c, d)$ in $[c_1, c_2] \times [d_{j+1}, d_j]$, let $p_2 = (c_2, d)$. Denote 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 $[c, c_2] \times [d, d_1]$ is the real infeasible region. Therefore, the approximation error induced by p_2 is smaller than that by p_1 , and the global optimal point would never be in a region with cost between c_1 and c_2 . ■

By Lemma 2, we just need to compute the local optimal point in the region $\mathbf{S}_{i,j}$ for all $i, j = 2, \dots, n-2$ such that $i \leq j$. We can easily compute that there are $N = \frac{(n-2)(n-3)}{2}$ regions in the real infeasible region defined by \mathcal{RP} . For example, in Fig. 4, we just need to consider the regions $\{\mathbf{S}_{2,2}, \mathbf{S}_{2,3}, \mathbf{S}_{3,3}\}$, where $n = 5$.

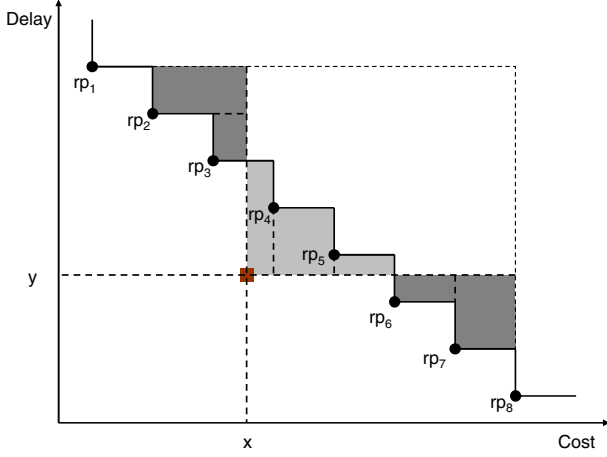


Fig. 5. An illustration for staircase approximation.

We first present a formula for the approximation error of a point in $\mathbf{S}_{i,j}$, where $i, j = 2, \dots, n-2$ and $i \leq j$. We then give a solution to find the local optimal point in this region that produces the smallest error. By computing the local optimal points in $\mathbf{S}_{i,j}$ for all $i, j = 2, \dots, n-2$ and $i \leq j$, we finally select the one with the minimum error as the global optimal point.

Given a point (x, y) in $\mathbf{S}_{i,j}$, we divide the real representative points into three groups based on x and y .

- 1) A set of representative points with cost smaller than x . As (x, y) is located in $\mathbf{S}_{i,j}$, $\mathcal{RP}_1 = \{rp_1, \dots, rp_i\}$.
- 2) A set of representative points with cost larger than x and delay larger than y , i.e., $\mathcal{RP}_2 = \{rp_{i+1}, \dots, rp_j\}$.
- 3) A set of representative points with delay smaller than y , i.e., $\mathcal{RP}_3 = \{rp_{j+1}, \dots, rp_n\}$.

For example, in Fig. 5, point (x, y) is located in the region $\mathbf{S}_{3,5}$. We thus have $\mathcal{RP}_1 = \{rp_1, rp_2, rp_3\}$, $\mathcal{RP}_2 = \{rp_4, rp_5\}$, and $\mathcal{RP}_3 = \{rp_6, rp_7, rp_8\}$. The approximated staircase is formed by rp_1 , (x, y) , and rp_8 . It is worth noting that the feasible region provided by \mathcal{RP}_2 is included in the feasible region defined by (x, y) . That is, (x, y) underestimates the delay in cost range of $[x, c_{j+1}]$. On the other hand, (x, y) overestimates the delay in ranges $[c_2, x]$ and $[c_{j+1}, c_n]$.

We first compute the feasible region provided by $\mathcal{RP}_1 = \{rp_1, \dots, rp_i\}$ but not included by the point (x, y) , which is part of overestimated area. As illustrated in Fig. 5, we divide this overestimated area into several subregions $R_{1,q} = [c_q, x] \times [d_q, d_{q-1}]$ for all $q = 2, \dots, i$. For example, in Fig. 5, we have $R_{1,2} = [c_2, x] \times [d_2, d_1]$ and $R_{1,3} = [c_3, x] \times [d_3, d_2]$. The formula for the overestimated areas, which are included in \mathcal{RP}_1 but not in (x, y) , is as follows.

$$\Delta_+^1(x, y) = \sum_{q=2}^i (d_{q-1} - d_q)(x - c_q). \quad (1)$$

Similarly, we can divide the overestimated area in cost range $[c_{j+1}, c_n]$ into several subregions $R_{2,q} = [c_q, c_{q+1}] \times [d_q, y]$ for all $q = j+1, \dots, n-1$. For each $R_{2,q}$. For example, in Fig. 5,

we have $R_{2,6} = [c_6, c_7] \times [d_6, y]$ and $R_{2,7} = [c_7, c_8] \times [d_7, y]$. The formula for the overestimated areas, which are included in \mathcal{RP}_3 but not in (x, y) , is

$$\Delta_+^2(x, y) = \sum_{q=j+1}^{n-1} (c_{q+1} - c_q)(y - d_q). \quad (2)$$

The total overestimated areas produced by (x, y) is

$$\Delta_+(x, y) = \Delta_+^1(x, y) + \Delta_+^2(x, y). \quad (3)$$

Based on a similar technique, we divide the underestimated area into several regions $R_{3,x} = [x, c_{i+1}] \times [y, d_i]$ and $R_{3,q} = [c_q, c_{q+1}] \times [y, d_q]$ for all $q = i+1, \dots, j$. For example, in Fig. 5, we have $R_{3,x} = [x, c_4] \times [y, d_3]$, $R_{3,4} = [c_4, c_5] \times [y, d_4]$, and $R_{3,5} = [c_5, c_6] \times [y, d_5]$. Formula (4) gives the size of the underestimated area.

$$\Delta_-(x, y) = (c_{i+1} - x)(d_i - y) + \sum_{k=i+1}^j (c_{k+1} - c_k)(d_k - y) \quad (4)$$

The total approximation error becomes

$$f_o(x, y) = \Delta_+((x, y)) + \Delta_-((x, y)). \quad (5)$$

We thus obtain a typical *nonlinear programming problem (NLP)* as expressed as follows:

$$\begin{aligned} \min f_o(x, y) \\ \text{s.t. } c_i \leq x \leq c_{i+1} \\ d_{j+1} \leq y \leq d_j \end{aligned} \quad (6)$$

Since the Hessian matrix of $f_o(x, y)$ is positive definite, we can use the Kuhn-Tucker method in [18] to compute the optimal values of x and y . We must consider the following cases:

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

In addition, we have

$$\begin{cases} \frac{\partial f_o(x, y)}{\partial x} = d_1 - 2d_i + y. \\ \frac{\partial f_o(x, y)}{\partial y} = c_n - 2c_{j+1} + x. \end{cases} \quad (7)$$

By (7), 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. Let $x = 2c_{j+1} - c_n$ and $y = 2d_i - d_1$. If $c_i \leq x \leq c_{i+1}$ and $d_{j+1} \leq y \leq d_j$, we obtain the local optimal point. If x does not fall between c_i and c_{i+1} , we let x be c_i or c_{i+1} . If y does not fall between d_{j+1} and d_j , we let y be d_j or d_{j+1} . Therefore, there are at most four possible local optimal points. We thus select the one with the minimum $f_o(x, y)$ as the local optimal point in $\mathbf{S}_{i,j}$.

We can see that the time for computing the local optimal point in the region $\mathbf{S}_{i,j}$ is constant. Since we need to compute $\frac{(n-2)(n-3)}{2}$ local optimal points, the time complexity of computing the global optimal point for a three-point approximated staircase is $\mathcal{O}(n^2)$, where n is the number of the optimal representative points.

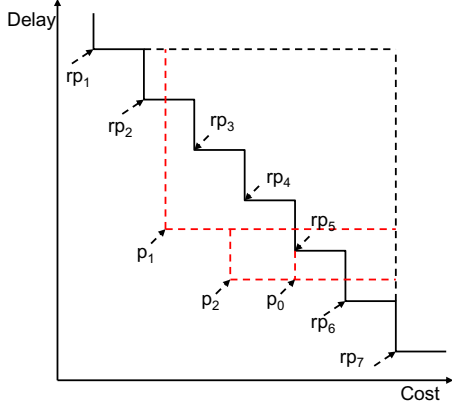


Fig. 6. Illustration for Lemma 3.

B. Multiple points approximation

In this section, we consider the case that $\mathcal{H} > 3$. We first prove the following lemma about two consecutive approximate points.

Lemma 3: Let ap and ap' be two consecutive approximate points where $ap.c < ap'.c$. If ap is in $\mathbf{S}_{i,j}$ and ap' is in $\mathbf{S}_{k,l}$, then $k \geq j + 1$.

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 $\{ap, p_0\}$. ■

For example, in Fig. 6, suppose that $ap = p_1$ and $ap' = p_2$. 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 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 difficult to obtain a general formulation representing the approximation error produced by \mathcal{AP} . By Lemma 3, each $\mathbf{S}_{i,j}$, where $i, j = 2, \dots, n-2$, contains at most one approximated point in \mathcal{AP} . We can 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 $N = \frac{(n-2)(n-3)}{2}$ be the number of the subregions, the number of combinations is $\binom{N}{\mathcal{H}-2}$. Therefore, we need to compute $\binom{N}{\mathcal{H}-2}$ local optimal \mathcal{AP} s 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

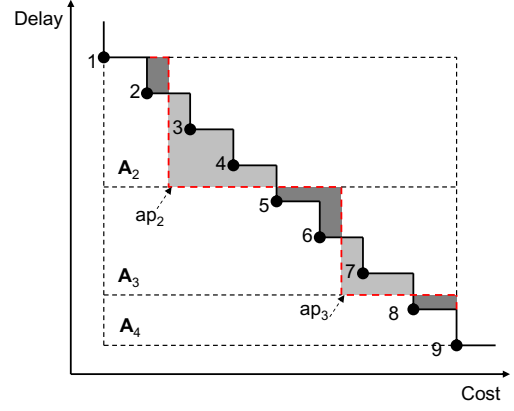


Fig. 7. An illustration for staircase approximation with multiple points.

combination.

Note that $ap_1 = rp_1$ and $ap_{\mathcal{H}} = rp_{\mathcal{H}}$, 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 $\mathbf{S}_{n-1, n-1} = [c_{n-1}, c_n] \times [d_n, d_{n-1}]$. 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 $i_m, j_m = 2, \dots, n-2$ and $m = 1, \dots, \mathcal{H}$. We have $j_m \geq i_m$ by Lemma 1 and $i_{m+1} > j_m$ by Lemma 3 for all $m = 2, \dots, \mathcal{H} - 1$. We also have $i_1 = j_1 = 1$ and $i_{\mathcal{H}} = j_{\mathcal{H}} = n - 1$.

For example, in Fig. 7, \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 and ap_3 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 $j_3 = 7$.

Let each point ap_m in \mathcal{AP} be represented by (x_m, y_m) , where $m = 1, \dots, \mathcal{H}$. Given a specific combination, in order to discuss how to compute the approximation error produced by \mathcal{AP} , we partition the region $[c_1, c_n] \times [d_n, d_1]$ based on delay into a set of subregions $\mathbf{A}_m = [c_1, c_n] \times [y_m, y_{m-1}]$ for all $m = 2, \dots, \mathcal{H}$. We then give the formula to compute the approximation error in each subregion. The total error is the sum of the error in each subregion. For example, in Fig. 7, we divide $[c_1, c_9] \times [d_9, d_1]$ into three subregions $\mathbf{A}_2 = [c_1, c_9] \times [y_2, d_1]$, $\mathbf{A}_3 = [c_1, c_9] \times [y_3, y_2]$, and $\mathbf{A}_4 = [c_1, c_9] \times [d_9, y_3]$. Now, we are going to discuss how to compute the error in each subregion \mathbf{A}_m .

The real representative points located in the region \mathbf{A}_m are $\mathcal{RP}_m = \{rp_{j_{m-1}+1}, \dots, rp_{j_m}\}$. For example, in Fig. 7, let $m = 2$, and we compute the approximation error in the region $\mathbf{A}_2 = [c_1, c_9] \times [y_2, d_1]$. Since $j_1 = 1$ and $j_2 = 4$, the set of representative points located in the region \mathbf{A}_2 is $\mathcal{RP}_2 = \{rp_2, rp_3, rp_4\}$. The set of representative points in the region \mathbf{A}_3 is $\mathcal{RP}_3 = \{rp_5, rp_6, rp_7\}$. The set of representative point in the region \mathbf{A}_4 is $\mathcal{RP}_4 = \{rp_8\}$.

In order to compute the approximation error in \mathbf{A}_m , we divide \mathcal{RP}_m into two subsets $\mathcal{RP}_{m,1} = \{rp_{j_{m-1}+1}, \dots, rp_{i_m}\}$ and $\mathcal{RP}_{m,2} = \{rp_{i_m+1}, \dots, rp_{j_m}\}$. 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} . The feasible region provided by $\mathcal{RP}_{m,2}$ must be included by (x_m, y_m) . In Fig. 7, $\mathcal{RP}_{2,1} = \{rp_2\}$ and

$\mathcal{RP}_{2,2} = \{rp_3, rp_4\}$. We also 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 $\mathcal{RP}_{\mathcal{H},2} = \emptyset$. For example, in Fig. 7, $\mathcal{RP}_{4,1} = \{rp_8\}$ and $\mathcal{RP}_{4,2} = \emptyset$.

The overestimated region in the region \mathbf{A}_m , denoted by Δ_+^m , is the feasible region provided by $\mathcal{RP}_{m,1}$, but not included by (x_m, y_m) . The computation method is similar to the computation of $\Delta_+^1(x, y)$ in (1). We give the formula of Δ_+^m as follows:

$$\Delta_+^m = (y_{m-1} - d_{j_{m-1}+1})(x_m - c_{j_{m-1}+1}) + \sum_{t=j_{m-1}+2}^{i_m} (d_{t-1} - d_t)(x_m - c_t). \quad (8)$$

The underestimated region in the region \mathbf{A}_m is denoted by Δ_-^m . The computation of Δ_-^m is similar to that of $\Delta_-(x, y)$ in (4). We give the formula of Δ_-^m as follows:

$$\Delta_-^m = (c_{i_{m+1}} - x_m)(d_{i_m} - y_m) + \sum_{t=i_{m+1}}^{j_m} (c_{t+1} - c_t)(d_t - y_m) \quad (9)$$

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

For example, in Fig. 7, let $m = 2$. We consider the approximation error in the region $[c_1, c_9] \times [y_2, d_1]$. Since $i_2 = 2$ and $j_1 = 1$, by (8), we obtain $\Delta_+^2 = (d_1 - d_2)(x_2 - c_2)$. By (9), 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. 7 are the total approximation error produced by \mathcal{AP} .

We thus obtain the nonlinear programming problem as follows:

$$\begin{aligned} \min f_o &= \sum_{m=2}^{\mathcal{H}} \Delta_+^m + \Delta_-^m \\ \text{s.t. } & c_{i_m} \leq x_m \leq c_{i_{m+1}} \\ & d_{j_{m+1}} \leq y_m \leq d_{j_m} \\ & m = 2, 3, \dots, \mathcal{H} - 1 \end{aligned} \quad (10)$$

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

$$\begin{cases} \frac{\partial f_o}{\partial x_m} = y_{m-1} + y_m - 2d_{i_m} \\ \frac{\partial f_o}{\partial y_m} = x_{m+1} + x_m - 2c_{j_{m+1}} \end{cases} \quad (11)$$

By (11), we can easily verify that the Hessian matrix of f_o is positive definite, so that we can use the Kuhn-Tucker method to find the optimal values of $\{(x_m, y_m), m = 2, \dots, \mathcal{H} - 1\}$. We let $\frac{\partial f_o}{\partial x_m} = 0$ and $\frac{\partial f_o}{\partial y_m} = 0$ for all $m = 2, \dots, \mathcal{H} - 1$. We thus get

$$\begin{cases} x_m = 2c_{j_{m+1}} - x_{m+1} \\ y_m = 2d_{i_m} - y_{m-1} \end{cases} \quad (12)$$

Since $x_{\mathcal{H}} = c_{\mathcal{H}}$ and $y_1 = d_1$, we can easily compute the values of x_m and y_m for all $m = 2, \dots, \mathcal{H} - 1$ based on (12).

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 $m = 2, \dots, \mathcal{H} - 1$, we have obtained the optimal values of \mathcal{AP} .

If $\exists m$ such that x_m is located outside 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 $k = m - 1, m - 2, \dots, 2$ based on the value of x_m . This process continues until all the values of $\{x_m, m = 2, \dots, \mathcal{H} - 1\}$ 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 $k = m + 1, \dots, \mathcal{H} - 1$. This process continues until all the values of $\{y_m, m = 2, \dots, \mathcal{H} - 1\}$ satisfy the corresponding constraints. Therefore, we will maximally obtain $2^{\mathcal{H}-1}$ possible local optimal \mathcal{AP} s, and we select the one with the minimum f_o .

Note that the time complexity of computing a local optimal \mathcal{AP} with a given specific combination is $2^{\mathcal{H}-1}$. It is expensive when \mathcal{H} is very large. \mathcal{H} also affects the space requirement in advertisement. Therefore, we should limit \mathcal{H} to be a small value. In this case, we still consider that the time complexity for computing a local optimal \mathcal{AP} is constant. Therefore, the time complexity for computing an optimal approximated multiple points staircase is $\binom{N}{\mathcal{H}-2} = \mathcal{O}(n^{2(\mathcal{H}-2)})$.

C. Discussion

Define $\mathcal{A}(b)$ as the border neighbor set of border node b . Each neighbor j of b advertises the supported QoS from itself to a destination, which is approximated by at most \mathcal{H} representative points. Moreover, b 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 b to a destination, via j , is approximated by at most \mathcal{H}^2 representative points. Since \mathcal{H} is predefined, we consider that the supported QoS from b to a destination, via its neighbor j , can be represented by a constant number of representative points. Therefore, the total supported QoS from b to a destination is thus approximated by $\mathcal{O}(|\mathcal{A}(b)|)$ representative points. Thus, the time complexity of our approach is $\mathcal{O}(|\mathcal{A}(b)|^{2(\mathcal{H}-2)})$. Since \mathcal{H} is a predefined small constant, our algorithm is polynomial with respect to the network topology.

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 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 of 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 a *crankbacked request*.

Moreover, the underestimation situation may cause the occurrence of the route oscillation, as referred to [6], [14]. The route oscillation makes our routing protocol diverge, we thus apply the *threshold checking* and *advertisement history checking* techniques [14] to suppress the advertisement. For a newly computed QoS, if several similar QoSes (say, two) have been advertised, this newly computed QoS will not be advertised. The comparison 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, given a threshold t , if for each newly computed representative point rp_{new} , there exists a old representative point rp_{old} such that $|rp_{new}.c - rp_{old}.c| \leq \min(rp_{new}.c, rp_{old}.c) \cdot t$ and $|rp_{new}.d - rp_{old}.d| \leq \min(rp_{new}.d, rp_{old}.d) \cdot t$, 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 a loop, it will drop this request.

IV. SIMULATION

In this section, we present our simulation results and compare the proposed topology QoS information aggregation method, called SCAM, with 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 [10]. The intradomain topology is generated based on the Waxman's model, and the interdomain topology is generated based on the Barabasi-Albert model. 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 $c_{i,j} \sim \text{uniform}[1, 100]$ and $d_{i,j} \sim \text{uniform}[1, 300]$.

In the networks, each domain contains 50 nodes, the average number of border nodes in each domain is four, and the average number of interdomain links is about four times the number of domains. 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 parameters is set to be 5%. Given the lower bound and upper bound of the cost constraint C_l and C_u , respectively, and the lower bound and the upper bound of the delay constraint D_l and D_u , respectively, we call the area $[C_l, C_u] \times [D_l, D_u]$ as the QoS request space. We have generated ten requests for each source-border-to-destination-domain pair with the cost requirement $c_{req} \sim \text{uniform}[C_l, C_u]$ and the delay requirement $d_{req} \sim \text{uniform}[D_l, D_u]$. 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 ratios: 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 by the routing protocol without topology aggregation. Therefore, the smaller Δ_+ , the greater the relative success ratio.

Crankback ratio [7] is defined as the ratio of the number of the crankbacked requests to the number of the accepted requests. There are two kinds of crankbacked requests: (1) an infeasible request accepted by the source node but is dropped by an intermediate node and (2) a feasible request for which the routing protocol cannot establish a feasible physical path.

Convergence speed refers to the time between the generation of the first advertisement message and the receipt of the last advertisement message in the network. *Advertisement overhead* is represented by the total number of the messages generated in the network.

C. Simulation results

In the simulation result figures, LSAM SR and SCAM SR are the approximated absolute success ratios produced by the algorithm LSAM and SCAM, respectively. LSAM RSR and SCAM RSR are the relative success ratios delivered by LSAM and SCAM, respectively. LSAM CR and SCAM CR are the crankback ratios produced by LSAM and SCAM, respectively.

Given a source and a destination domain, define (c_l, d_l) and (c_u, d_u) as the QoS parameter 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. 8. 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. 9. 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 close to 1 and the crankback ratio is less than 7%, no matter what the actual absolute success ratio is. However, the performance of LSAM changes obviously as the QoS request space changes. For instance, in Fig. 9(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. 9(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 sum of

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.

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. 10 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 errors produced by our approach and LSAM changes slowly as the network size increases, which demonstrates that the proposed aggregation method is suitable for 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 converges faster than LSAM, and the number of the messages generated by our approach is smaller than that by LSAM. 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.

V. CONCLUSION

In this paper, we investigate a very challenging problem, supporting QoS routing in the Internet. The challenge, scala-

bility, 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 simulations, we show that our approach can serve the requests with the QoS requirements 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 and the QoS request space do not have much affect the performance of our algorithm, which demonstrates that our algorithm is scalable and stable. On the other hand, 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, "An approximation algorithm for QoS routing with two additive constraints," *IEEE ICNP 2008*, pp. 328-337, October 2008.
- [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 networks," *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.-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.
- [8] W. Lee, "Spanning tree method for link state aggregation in large communication networks," in *Proc. IEEE INFOCOM*, pp. 297-302, 1995.
- [9] W. Lee, "Topology aggregation for hierarchical routing in ATM networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 25, pp. 82-92, April 1995.
- [10] 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.
- [11] I. T. Okumus, H. A. Mantar, J. Hwang, and S. J. Chapin, "Inter-Domain QoS Routing on Diff'serv Networks: A Region-Based Approach," *Computer Communications*, vol. 28, no. 2, pp. 174-188, February 2005.
- [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] Y. Tang, S. Chen, Y. Ling, "State aggregation of large network domains," *Computer Communications*, vol. 30, no. 4, pp. 873-885, February 2007.
- [17] 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.

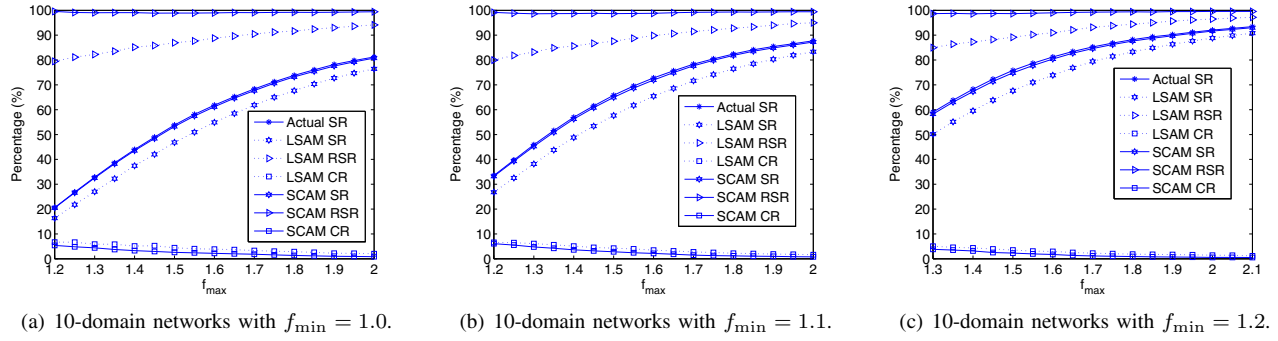


Fig. 8. The performance of the interdomain algorithms by changing f_{\max} .

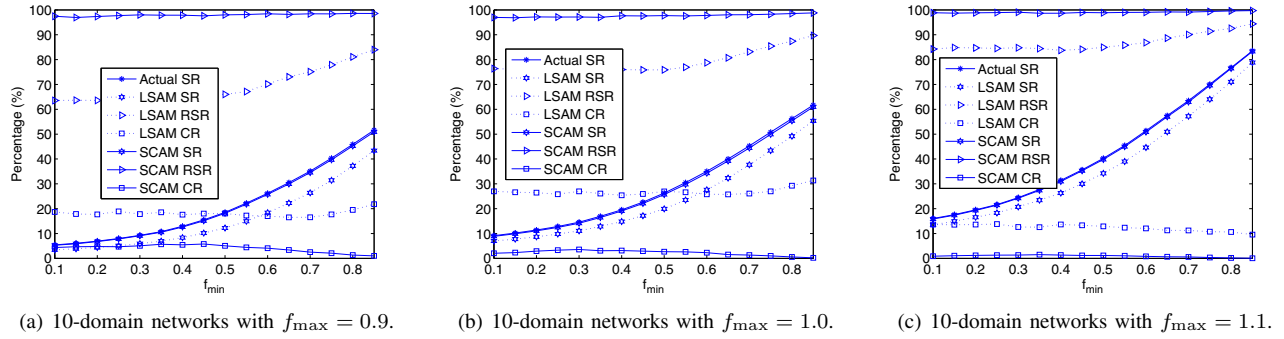


Fig. 9. The performance of the interdomain algorithms by changing f_{\min} .

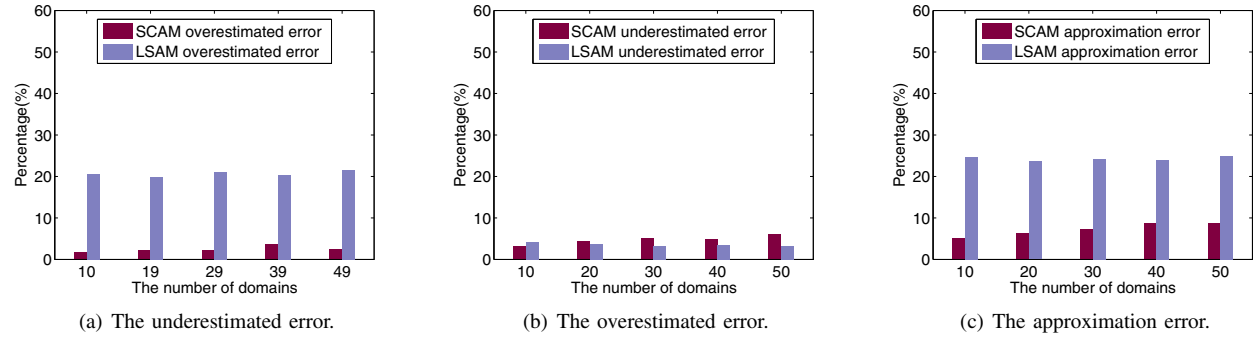


Fig. 10. Comparing the performance of approximation error.

- [18] W. L. Winston, *Operations Research: Applications and Algorithms*. Duxbury Press; 4 edition, July, 2003.
- [19] 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.
- [20] 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.
- [21] R. Hou, K.-S. Lui, K.-C. Leung, and F. Baker, "QoS routing with topology aggregation in networks," *Technical Report (TR-2008-002)*, Department of Electrical and Electronical Engineering, The University of Hong Kong, <http://www.eee.hku.hk/research/research-reports.htm>, August 2008.