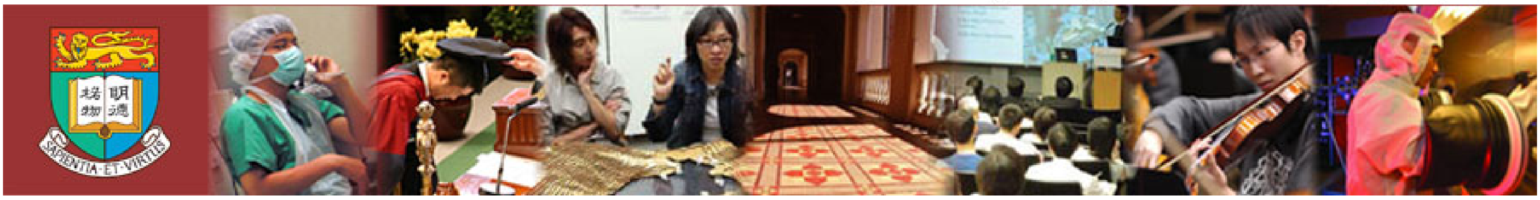


This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Quality-of-Service Routing with Two Concave Constraints

Ka-Chung Leung, King-Shan Lui, Ka-Cheong Leung
Dept. of Electrical and Electronic Eng., Univ. of Hong Kong
E-mail: {kachung, kslui, kcleung}@eee.hku.hk

Fred Baker
Cisco Research Center, USA
E-mail: fred@cisco.com

***Abstract*—Routing is a process of finding a network path from a source node to a destination node. A good routing protocol should find the "best path" from a source to a destination. When there are independent constraints to be considered, the "best path" is not well-defined. In our previous work, we developed a line segment representation for Quality-of-Service routing with bandwidth and delay requirements. In this paper, we propose how to adopt the line segment when a request has two concave constraints. We have developed a series of operations for constructing routing tables under the distance-vector protocol. We evaluate the performance through extensive simulations.** [1]

I. INTRODUCTION

Quality-of-Service (QoS) Routing refers to the process of identifying a path that satisfies certain requirements such as bandwidth, delay, security level, cost, reliability, etc. A good routing protocol should find the "best path" from a source to a destination. In most standard routing protocols, the "best path" usually refers to the shortest path in terms of a single metric. This single metric may be the number of hops, delay, bandwidth, or a fixed formula combining a vector metric to one value for use in the routing algorithm as in Interior Gateway Routing Protocol (IGRP), and so on. Unfortunately, when it comes to two or more metrics, the problem of *finding the best path* is not trivial anymore.

In this paper, we consider two simultaneous *concave* metrics. A concave metric example is bandwidth (or capacity). The bandwidth of a path is the minimum bandwidth among the links on the path. We can define different bandwidth metrics for a link. For example, a link can have a certain maximum capacity but due to traffic dynamic, its available capacity fluctuates. For multimedia traffic, we may want to identify a path with certain maximum and average capacities. Our protocol works with any two arbitrary concave metrics and we denote them as Metric S and Metric W in our discussion.

Consider the simple network in Figure 1 where tuple (x, y) associated with each edge represents the QoS metrics of S and W, respectively. Note that the network contains both unidirectional edges and bidirectional edges. The QoS of the path $A \to E \to D$ is (min(13, 6), min(10, 11)) = (6, 10) while the QoS of path $A \to B \to C \to D$ is (9, 7). The former is better in terms of W and the latter is better in terms of S. No matter which path is selected as the "best" and kept in the routing table, some feasible QoS requests will not be admitted. Suppose that Node A decides to keep path $A \to B \to C \to D$ in its routing table and then receives a routing request to D that requires 8 units of bandwidth. By checking its routing table, A thinks that there is no path from itself to D having 8 units of bandwidth and rejects the request. However, the request is actually feasible since it is supported by path $A \to E \to D$.

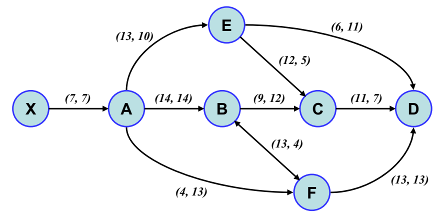
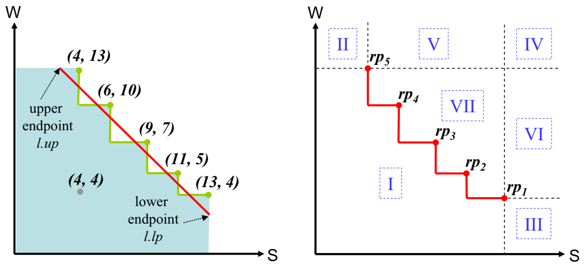


Fig. 1. A simple network where (x, y) represents two concave QoS metrics.

If the parameters of all the paths from a source to a destination are plotted on the Cartesian plane with W on the y-axis and S on the x-axis, the region of supported services forms a staircase. Figure 2(a) illustrates the idea. Points $(4, 4)$, $(4, 13)$, $(6, 10)$, $(9, 7)$, $(11, 5)$, and $(13, 4)$ refer to the parameters of the paths from Node A to D. $(9, 7)$ is definitely better than $(4, 4)$ since it is better in both metrics W and S. We say that $(9, 7)$ is more *representative* than $(4, 4)$. Geographically, the area spanned by $(4, 4)$ is a subset of the area spanned by $(9, 7)$. However, $(9, 7)$ is neither better than $(6, 10)$ nor $(11, 5)$. The shaded area represents the feasible requests that can be supported by at least one path. To see why the area is a staircase, we look at the region supported by a certain path, say $(9, 7)$. $(9, 7)$ can support all requests that requiring S not larger than 9 units and that W not larger than 7 units. The supported S and W values of that path fall into the lower left quadrant of $(9, 7)$. The region supported by all paths is the union of the regions of all paths, forming a staircase as shown in Figure 2(a). This staircase can be represented by the points on the convex corners of the stairs and these points are referred to as *representative points* that can be identified in polynomial time [1].

Although the region of supported services can be represented by a set of representative points, it is not scalable to advertise the whole staircase to neighbors for routing. For example, in Figure 1 with the distance-vector routing protocol, A should advertise to X the QoS information from itself to D. It would be too expensive if the whole staircase is advertised. We have developed a *line segment* approach to solve the problem [1]. A line, as shown in Figure 2(a), is used to approximate a staircase. Since a line can be uniquely represented by two points, advertising line segments reduces

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



(a) Parameters on W-S plane. (b) Seven regions on the W-S plane.

Fig. 2. QoS Routing with two concave constraints.

the size of the disseminated information dramatically. We further studied how to apply the representation for bandwidth and delay constraints in the distance-vector protocol [2]. In this paper, we study how to apply the line segment representation when both QoS metrics are concave.

QoS routing has been studied for many years. Earlier works mainly focused on how to identify a path that satisifies multiple constraints in a flat network with global topology information [3], [4]. Sobrinho studied the issues of QoS routing in the Internet where packets are forwarded in a hop-by-hop manner [4]. To carry out and deploy QoS in the inter-domain or inter-provider domain, IntServ[5] and DiffServ[6] are the two major proposals developed by the IETF. However, routing is not addressed. RFC2676 [7] describes how to extend OSPF to support QoS. Both route computation and OSPF modifications are discussed. Recently, the overlay approach is proposed for supporting interdomain QoS routing. Different QoS planes are built among domains and routes are identified accordingly. The QRON approach in [8] provides an overlay solution to interdomain QoS routing by means of a source-based, hierarchical, link-state protocol. The OverQoS approach in [9] is another overlay in which the existence of pre-determined set of paths is assumed and no path selection algorithm is provided in this respect. The authors in [10] proposes a framework to combine the Border Gateway Protocol (BGP) extended with traffic engineering and an overlay approach. Another overlay approach in [11] uses a link-state and source-specified QoS routing architecture for the interdomain routing. The MESCAL approach [12] presents an architecture for supporting interdomain QoS across the Internet. They focus on a business model and a framework to provide interdomain QoS routing. The simulation result of the architecture is presented in [13]. To the best of our knowledge, our work is the first one that considers inter-domain routing with two concave metrics.

II. Network Model and Problem Statement

A simple network is shown in Figure 3. A *border node*, denoted as a black node in the figure, is a node that connects to other domain. To simplify our discussion, we refer border node i of domain d as $d.i$. In Figure 3, $A.1$ and $A.2$ are border nodes of domain A.

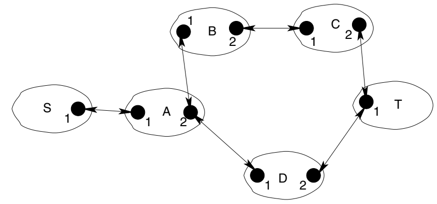


Fig. 3. A simple two-level hierarchical network.

The metric of a link is expressed as an ordered set of uncorrelated metric values (s, w). Each (s, w) represents a single point on the W-S plane. A *physical path* from node v_0 to node v_k, which is denoted by $(v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_{k-1} \rightarrow v_k)$, consists of a set of directed links (v_i, v_{i+1}), for $0 \leq i < k$. Let $(s_{i\rightarrow i+1}, w_{i\rightarrow i+1})$ be the ordered set of link (v_i, v_{i+1}). Value of metric S of the path from v_0 to v_k is $\min_{i=0}^{k-1}\{s_{i\rightarrow i+1}\}$ while the value of W is $\min_{i=0}^{k-1}\{w_{i\rightarrow i+1}\}$. For example, if $k = 3$ and the parameters of (v_0, v_1), (v_1, v_2), and (v_2, v_3) are (3,5), (5,4), and (6,4), respectively, then the S value of $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_3$ is $\min\{3, 5, 6\} = 3$ and the value of W is $\min\{5, 4, 4\} = 4$. In the geometrical representation, like in Figure 2(a), the QoS parameter pair of a physical path is denoted by a point in the W-S plane.

A BGP border router advertises its path information to neighboring border routers to build routing tables. For example, $C.2$ knows that $T.1$ is its direct neighbor and it relays this information to $C.1$. When $C.1$ gets the information from $C.2$, $C.1$ knows that it can reach $T.1$ through $C.2$. It then informs $B.2$ that it can reach $T.1$. If a single QoS parameter is considered, say bandwidth, border nodes should also advertise the bandwidth information. There are two paths from $A.2$ to $T.1$. Suppose that the bandwidth value of the lower path is larger. Then, $A.2$ should specify in its routing table that the next hop neighbor leading to $T.1$ is $D.1$. Furthermore, $A.2$ advertises only the lower path's bandwidth value to $A.1$. However, when two (QoS) parameters are considered, the distance-vector calculation and advertisement procedure are not trivial. In this paper, we identify and solve the three main problems that follow from routing with two concave metrics:

1) How can $B.1$ find the QoS from itself to $T.1$? We assume that the QoS from $C.1$ to $C.2$ and the QoS from $B.1$ to $B.2$ are represented by line segments as discussed in Section I. Although the QoS of the inter-domain links $C.2$ to $T.1$ and $B.2$ to $C.1$ are both (s, w) tuples, we have to join these QoS parameters together to find the QoS from $B.1$ to $T.1$. This is described in Section III-A.
2) Also, when $A.2$ advertises to $A.1$ the QoS information leading to T, it may not be possible to determine whether the upper path or the lower path is better. Therefore, in order to embed as much of the underlying QoS information of the paths as possible in advertisement, we have to *aggregate* the QoS of the two paths together. This is described in Section III-B.
3) Finally, how should the construction of routing tables be changed to cope with the new information? Due to space

limitation, we refer readers to our technical report [14] for the solution.

III. QoS Routing Mechanism

A. QoS Join Operation (denoted as $\oplus$)

A logical inter-domain or intra-domain path $(v_x \rightarrow v_y)$ can be a QoS point or a QoS line. For example, referring to Figure 3, the QoS from $S.1$ to $A.1$ is a QoS point (a single point on the W-S plane) since this path consists of one link. On the other hand, the QoS from $A.2$ to $T.1$ can be a QoS line as shown in Figure 2(a) since there are multiple paths from $A.2$ to $T.1$. To find the join of two paths $(v_a \rightarrow v_b)$ and $(v_b \rightarrow v_c)$, three cases are possible — a point joining another point, a point joining a line, and a line joining another line. For the case of a point rp_1 joining another point rp_2 $(rp_1 \oplus rp_2)$, the join result is a QoS point at $(min(rp_1.s, rp_2.s), min(rp_1.w, rp_2.w))$. In the following subsections, we prove and describe the join operations of a point against a line and a line against another line.

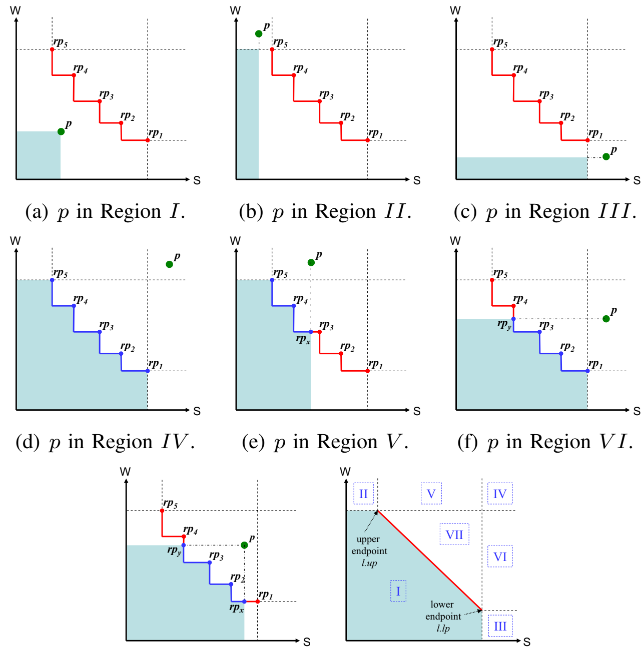
1) Joining a point and a line: To provide a joining mechanism of a QoS point p and a line l, we shall derive it by considering the original staircase $stair$ with representative points $rp_1, rp_2, ..., rp_n$ that are approximated by the line l. Assume p is the QoS of the link $(v_a \rightarrow v_b)$ and $stair$ is QoS of the path $(v_b \rightarrow v_c)$. The join operation can be viewed as combining the QoS of p with every rp_i in $stair$. In other words, the QoS of the path $(v_a \rightarrow v_b \rightarrow v_c)$ is the union of the individual $p \oplus rp_i$. Therefore, we have:

$$\begin{aligned} p \oplus stair &= (p \oplus rp_1) \cup (p \oplus rp_2) \cup ... \cup (p \oplus rp_n) \\ &= \bigcup_{i=1}^{n} (min(p.s, rp_i.s), min(p.w, rp_i.w)) \quad (1) \end{aligned}$$

One property of joining links with a concave metric is that the QoS will never increase by adding more links. Thus, $p \oplus stair$ must cover an area no larger than that of p. Given a staircase $stair$ on the W-S plane, the plane can be divided into seven regions, as shown in Figure 2(b). The mechanism of the joining is different when point p is located in different regions on the plane. Mathematically, p is said to be in a particular region according to the following:

- *Region I:* There exists some points $rp_z \in stair$ such that rp_z is more representative than p.
- *Region II:* $p.s < rp_n.s$ and $p.w \geq rp_n.w$.
- *Region III:* $p.s \geq rp_1.s$ and $p.w < rp_1.w$.
- *Region IV:* $p.s \geq rp_1.s$ and $p.w \geq rp_n.w$.
- *Region V:* $rp_n.s \leq p.s < rp_1.s$ and $p.w \geq rp_n.w$.
- *Region VI:* $p.s \geq rp_1.s$ and $rp_1.w \leq p.w < rp_n.w$.
- *Region VII:* $p.s < rp_1.s$ and $p.w < rp_n.w$ and there does not exist any point $rp_z \in stair$ such that rp_z is more representative than p.

Joining for p in each region is illustrated in Figure 4. Due to space limitation, we only describe Regions I-III. We refer readers to our technical report [14] for the details of the other regions.



(a) p in Region I. (b) p in Region II. (c) p in Region III. (d) p in Region IV. (e) p in Region V. (f) p in Region VI. (g) p in Region VII. (h) Region partitioned by a line.

Fig. 4. Joining of a QoS point and a QoS stair

1) *Region I:* rp_z is more representative than p. Therefore, $rp_z \oplus p = p$. This is also the best we can have. The resulting QoS for the join operation is shown in Figure 4(a).
2) *Region II:* For each rp_i on $stair$, $p \oplus rp_i = (p.s, rp_i.w)$. As $rp_n.w \geq rp_i.w$, $p \oplus stair = (p.s, rp_n.w)$ as in Figure 4(b).
3) *Region III:* Based on a similar argument as in Region II, $p \oplus stair = (rp_1.s, p.w)$ as in Figure 4(c).

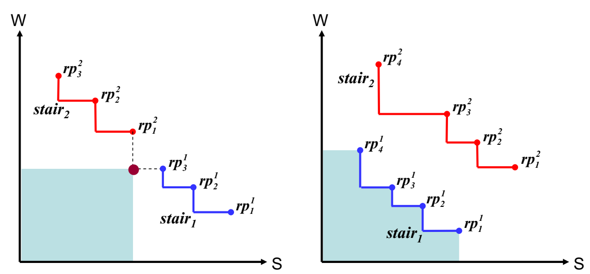
We have proved the result of joining for p being located in different regions of $stair$. However, in real situation, a line l used to approximate the staircase is advertised to a neighbor instead of the staircase itself. The line l is obtained by applying linear regression to all representative points, and this operation requires a time complexity of $O(n)$ where n is the number of such points. With linear regression, we have to approximate rp_1 with the lower endpoint of l, $l.lp$, and rp_n with the upper endpoint of l, $l.up$. $l.lp$ may have its W value different from that of rp_1 while $l.up$ may have its S value different from that of rp_n. Also, we have to believe that there are infinite number of representative points along l, so the indicated service area will be different from the area covered by the original staircase. In this way, Figure 2(b) has to be modified to deal with such approximations, as redrawn in Figure 4(h).

2) Joining a line and another line: Similar to the previous subsection, to provide a joining mechanism of a QoS line l_1 and another line l_2, we shall derive it by considering the original staircase $stair_1$ with representative points $rp^1_1, rp^1_2, ..., rp^1_n$ and $stair_2$ with representative points $rp^2_1, rp^2_2, ..., rp^2_n$ that are approximated by the lines l_1 and l_2, respectively.

Assume that $stair_1$ and $stair_2$ are the QoSs of the paths $(v_a \rightarrow v_b)$ and $(v_b \rightarrow v_c)$, respectively. As every rp_i^1 (where $1 \leq i \leq n_1$) combined with every rp_j^2 (where $1 \leq j \leq n_2$) represents the path from v_a to v_c via v_b, the join operation can be viewed as combining the QoS of every rp_i^1 on $stair_1$ with every rp_j^2 on $stair_2$. By taking union of all the service areas that are supported by different paths, the QoS of the path $(v_a \rightarrow v_b \rightarrow v_c)$ can be found as:

$$stair_1 \oplus stair_2 = \bigcup_{i=1}^{n_1} (rp_i^1 \oplus stair_2) \qquad (2)$$

Then, we can enumerate every element $(rp_i^1 \oplus stair_2)$ in Equation 2 using Equation 1 and finally obtain the union of n^2 tuples.



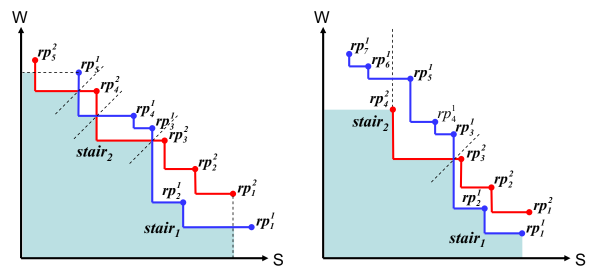
(a) Staircases in disjoint ranges. (b) $stair_1$ in Region I of $stair_2$.

Fig. 5. Two fundamental cases of $stair_1$ and $stair_2$.

To find the join result of arbitrary $stair_1$ and $stair_2$, we first need to define and formulate the join result of two fundamental cases, as shown in Figure 5. Case (1) is when $stair_1$ and $stair_2$ have disjoint S and W ranges with $rp_{n_1}^1.s > rp_1^2.s$. Case (2) is when every representative point in $stair_1$ is in Region I of $stair_2$. With these two fundamental cases, we will be able to derive the join result of any geometrical placement of $stair_1$ and $stair_2$.

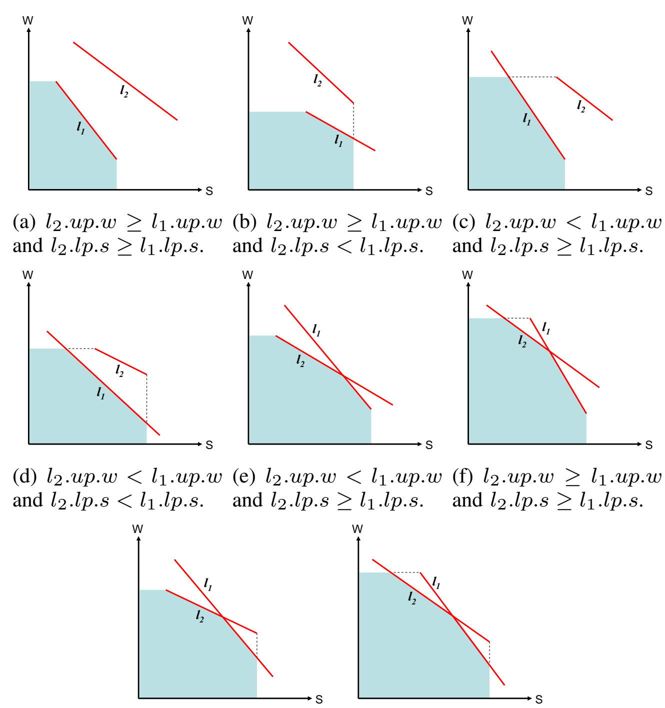
- Fundamental case (1):
  In $stair_1$, all representative points are positioned in Region III of $stair_2$. Applying the result of Section III-A.1, the joining of every rp_i^1 (where $1 \leq i \leq n_1$) with $stair_2$ results in a single point at $(rp_1^2.s, rp_i^1.w)$. Since the point $(rp_1^2.s, rp_{n_1}^1.w)$ has the largest W value, it must be the only representative point that is formed by $stair_1 \oplus stair_2$. This case is shown in Figure 5(a).
- Fundamental Case (2):
  When every representative point in $stair_1$ is in Region I of $stair_2$, the joining of every rp_i^1 (where $1 \leq i \leq n_1$) with $stair_2$ results in the original point at $(rp_i^1.s, rp_i^1.w)$ according to the result from Section III-A.1. Finally, the original staircase $stair_1$ will remain after the joining. This case is shown in Figure 5(b).

With the fundamental cases, we can return to the general situation. The approach is like this. Given any $stair_1$ and $stair_2$, if they entirely belong to one of the above fundamental cases, the derivation is finished. Otherwise, we first need to break the staircases into segments. Whenever there is an intersection point between $stair_1$ and $stair_2$, the two staircases are being segmented by a breaking line. Two examples of segmentation are shown in Figure 6. Then, we separately join every segment in $stair_1$ to $stair_2$ to obtain the join result. Figure 7 illustrates the different results of $l_1 \oplus l_2$, where l_1 and l_2 are approximations of $stair_1$ and $stair_2$, respectively.



(a) Illustration 1. (b) Illustration 2.

Fig. 6. Segmentation of $stair_1$ and $stair_2$.



(a) $l_2.up.w \geq l_1.up.w$ and $l_2.lp.s \geq l_1.lp.s$. (b) $l_2.up.w \geq l_1.up.w$ and $l_2.lp.s < l_1.lp.s$. (c) $l_2.up.w < l_1.up.w$ and $l_2.lp.s \geq l_1.lp.s$.

(d) $l_2.up.w < l_1.up.w$ and $l_2.lp.s < l_1.lp.s$. (e) $l_2.up.w < l_1.up.w$ and $l_2.lp.s \geq l_1.lp.s$. (f) $l_2.up.w \geq l_1.up.w$ and $l_2.lp.s \geq l_1.lp.s$.

(g) $l_2.up.w < l_1.up.w$ and $l_2.lp.s < l_1.lp.s$. (h) $l_2.up.w \geq l_1.up.w$ and $l_2.lp.s < l_1.lp.s$.

Fig. 7. Graphical representation of joining l_1 and l_2, with (e)-(h) having line intersections (aggregated service area denoted as the shaded region).

B. Line Aggregation

In order to enable the use of QoS line segment in distance-vector routing, another issue we need to consider is the mechanism for a node to determine how it should advertise the QoSs of several paths. Consider the case of Node $A.2$ in Figure 3. The node knows the QoSs of the two paths leading to T, the upper path $A.2 \rightarrow B \rightarrow C \rightarrow T$ and the lower

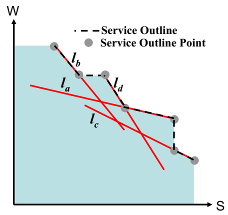


Fig. 8. Line aggregation mechanism.

path $A.2 \rightarrow D \rightarrow T$, and now it is its turn to advertise this information to its neighbor $A.1$. Which QoS should $A.2$ advertise? If one QoS is better than the other, $A.2$ can simply tell $A.1$ the better QoS.

However, if the QoSs that a node possesses are the ones shown in Figure 8, it is difficult to tell which one is "better". Our approach to the problem is to *aggregate* the QoSs of the paths, representing the aggregated QoS with one new line segment. By doing so, we aim to embed as much of the QoS information as possible into one line segment.

In the W-S plane, the QoS supported by an arbitrary line segment is the lower-left quadrant of the line. When a node receives multiple line segments for the same destination, the aggregated QoS is the union of the services supported by those lines. This is illustrated by the shaded area in Figure 8. In order to embed as much of the QoS information as possible, a node should advertise the aggregated QoS instead of broadcasting one of the QoSs only. As shown in the figure, such service outline in general does not form a straight line but a polyline. The QoS polyline can be uniquely identified by specifying its service outline points (i.e., the dots in Figure 8). As the number of service outline points grows linearly with the number of QoS lines to the same destination, advertising all service outline points is not scalable in the Internet. In order to solve this problem, we again use linear regression on the service outline points to approximate the service outline of the aggregated QoS with one line segment, which leads to a tradeoff between scalability and accuracy in route information. The operation of finding the service outline points requires a time complexity of $O(m^2)$ while obtaining the service outline using linear regression requires $O(m)$ time, where m is the number of QoS line segments.

IV. Performance Evaluation and Conclusion

In this section, we present the performance evaluation of our protocol through simulation. We evaluated the protocol performance using a self-written C++ network simulator, with the network topologies generated by the Boston University Representative Internet Topology Generator (BRITE). The integer S and W values of links are randomly picked from [5, 10]. The simulator performs both intra-domain and inter-domain routings with two-level hierarchical networks so as to emulate Internet routing. At this stage, we simulate a static network with no link cost changes throughout route information exchange process. After then, numerous QoS requests are being served. There are 2,500 requests for each network. The integer S and W requirements are randomly picked from [5, 9]. By launching simulations on various network topologies, the routing accuracy of our protocol and how well it is serving various QoS requests were investigated.

We measure performance in terms of *success ratio* and *crankback ratio*. Success ratio is the total number of accepted feasible requests divided by the total number of feasible requests. It measures how well our routing protocol is in serving feasible requests. Crankback ratio is defined as the ratio of accepted requests that cannot be served. It is easy to see that a good QoS routing protocol should have high success ratio and low crankback ratio.

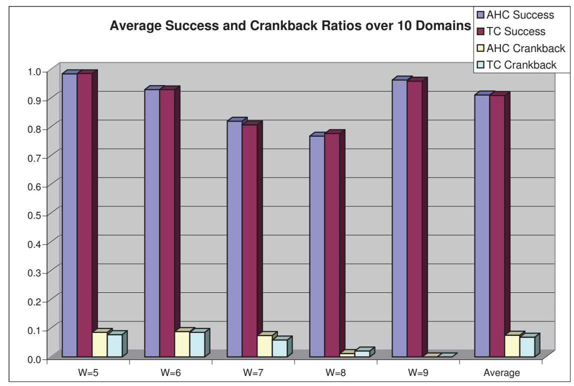


Fig. 9. Simulation results for QoS routing with two concave constraints.

The simulation results are shown in Figure 9. We simulated 10 different networks using both threshold checking (TC) and advertisement history checking (AHC) modes [2], and all results shown below are values averaged over 10 domains. From the simulation, the average success ratio is over 90% for both TC and AHC modes, showing that the protocol is good at estimating the actual QoS of the path towards the destination. The average crankback ratio is about 7% for both TC and AHC modes, which is formed by inaccuracies introduced in the route information exchange process. In conclusion, the simulation results show that our protocol is very efficient.

References

[1] K.-S. Lui, K. Nahrstedt, and S. Chen, "Routing with Topology Aggregation in Delay-Bandwidth Sensitive Networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 1, pp. 17–29, February 2004.
[2] W.-Y. Tam, K.-S. Lui, S. Uludag, and K. Nahrstedt, "Quality-of-Service Routing with Path Information Aggregation," *Elsevier Computer Networks Journal*, vol. 51, no. 12, pp. 3574–3594, August 2007.
[3] F. Kuipers, P. Van Mieghem, T. Korkmaz, and M. Krunz, "An Overview of Constraint-based Path Selection Algorithms for QoS Routing," *IEEE Communications Magazine*, pp. 50–55, December 2002.
[4] Y. Xiao, K. Thulasiraman, and G. Xue, "QoS Routing in Communication Networks: Approximation Algorithms Based on the Primal Simplex Method of Linear Programming," *IEEE Transactions on Computers*, pp. 815–829, July 2006.
[5] R. Braden, D. Clark, and S. Shenker, "RFC 1633: Integrated Services in the Internet Architecture: an Overview," December 2000.
[6] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC 2475: An Architecture for Differentiated Services," February 2003.
[7] G. Apostolopoulos et. al, "RFC 2676: QoS Routing Mechanisms and OSPF Extensions," August 1999.
[8] Z. Li and P. Mohapatra, "QRON: QoS-Aware Routing in Overlay Networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 29–40, January 2004.
[9] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz, "OverQoS: An Overlay Based Architecture for Enhancing Internet QoS," pp. 71–84, March 2004.
[10] M. Yannuzzi, A. Fonte, X. Masip-Bruin, E. Monteiro, S. Sánchez-López, M. Curado, and J. Domingo-Pascual, "A Proposal for Inter-domain QoS Routing Based on Distributed Overlay Entities and QBGP," pp. 257–267, September 2004.
[11] I. T. Okumus, H. A. Mantar, J. Hwang, and S. J. Chapin, "Inter-Domain QoS Routing on Diffserv Networks: A Region-Based Approach," *Computer Communications*, vol. 28, no. 2, pp. 174–188, February 2005.
[12] M. P. Howarth, P. Flegkas, G. Pavlou, N. Wang, P. Trimintzios, D. Griffin, J. Griem, M. Boudcadair, A. Asgari, and P. Georgatsos, "Provisioning for Interdomain Quality of Service: the MESCAL Approach," *IEEE Communications Magazine*, pp. 129–137, June 2005.
[13] D. Griffin, J. Spenser, J. Griem, M. Boucadair, P. Morand, M. Howarth, N. Wang, G. Pavlou, A. Asgari, and P. Georgatsos, "Interdomain Routing through QoS-Class Planes," *IEEE Communications Magazine*, pp. 88–95, February 2007.
[14] K.-C. Leung, K.-S. Lui, and K.-C. Leung, "Quality-of-Service Routing with Two Concave Constraints," *Technical Report, Department of Electrical and Electronic Engineering, The University of Hong Kong*, September 2007.