Performance analysis of quantization-based approximation algorithms for precomputing the supported QoS ☆

Ronghui Hou [a,*], King-Shan Lui [b], Ka-Cheong Leung [b], Fred Baker [c]

[a] The State Key Lab of Integrated Service Networks, Xidian University, China
[b] Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong
[c] Cisco Research Center, San Jose, CA 95134, USA

ABSTRACT

Precomputation of the supported QoS is very important for internet routing. By constructing routing tables before a request arrives, a packet can be forwarded with a simple table lookup. When the QoS information is provided, a node can immediately know whether a certain request can be supported without launching the path finding process. Unfortunately, as the problem of finding a route satisfying two additive constraints is NP-complete, the supported QoS information can only be approximated using a polynomial time mechanism. A good approximation scheme should reduce the error in estimating the actual supported QoS. Nevertheless, existing approaches which determine this error may not truly reflect the performance on admission control, meaning whether a request can be correctly classified as feasible or infeasible. In this paper, we propose using a novel metric, known as *distortion area*, to evaluate the performance of precomputing the supported QoS. We then analyze the performance of the class of algorithms that approximate the supported QoS through discretizing link metrics. We demonstrate how the performance of these schemes can be enhanced without increasing complexity. Our results serve as a guideline on developing discretization-based approximation algorithms.

© 2013 The Authors. Published by Elsevier Ltd. All rights reserved.

1. Introduction

As the demand for deploying real-time and multimedia applications over the internet is increasing, providing guaranteed quality-of-service (QoS) for these applications becomes more and more important. In general, the QoS requirements can be divided into two categories: *bottleneck* metric and *additive* metric. The additive metric of a path is the sum of the metrics of the links along the path, while the bottleneck metric of a path is the minimum value of the metrics of the links along the path. For example, bandwidth is a *bottleneck* QoS metric, while delay and delay jitter are *additive* QoS metrics. In this work, we consider connection requests that have two additive QoS requirements or constraints, such as in delay and cost. To simplify our discussion, we assume that delay and cost are the two additive metrics under consideration, although our analysis and method can be applied to any additive metrics.

Many existing works study how to identify a feasible path for a request with two additive constraints, which is an NP-complete

problem. These works usually assume either the cost or the delay requirement which is known. Nevertheless, such reactive routing mechanism, which finds a path after the requirement is known, cannot provide enough information to support efficient admission control. When a request is received, a node cannot immediately tell whether a possible feasible path exists until a path finding process is launched based on the requested cost/delay. On the other hand, by precomputing the supported QoS information, a source can immediately determine whether the connection request can be supported by the network. Moreover, accepting a new connection will not violate the service guarantees for the existing traffics, and also the transmission route satisfies the QoS requirement of the new connection.

The problem of computing the supported QoS between two nodes is more complicated than the extensively studied *multi-constrained path (MCP)* problem or the *delay-constrained least cost (DCLC)* path problem. The DCLC problem is also called the *restricted shortest path (RSP)* problem. The RSP problem aims at finding the minimum delay path among the paths that satisfy a certain cost constraint. The MCP problem studies finding a path satisfying both specified cost and delay constraints. Both problems focus on finding a single path between two nodes with a given (cost) constraint. Our problem, also known as the *all-costs optimal path (ACOP)* problem (Orda and Sprintson, 2003), finds, for each cost c, a c-cost constrained path from a source to a destination with the minimum delay. In other words, instead of finding a

single path given a cost constraint, the ACOP problem aims at finding a set of paths representing the supported QoS.

Due to the NP-complete nature of the problem, some approximation mechanisms have been developed (Garroppo et al., 2010). They usually identify a path with a cost (or delay) within a certain deviation from the optimal one. Denote c as the estimated optimal cost of all the paths satisfying a given delay constraint d_0, which is computed by an algorithm, and c_{opt} as the optimal cost of all the paths satisfying the delay constraint d_0 in the network. $c - c_{opt}$ is thus called the *cost deviation* at the delay constraint of d_0. An algorithm is "better" if the deviation is smaller. While deviation is appropriate for measuring the performance of the DCLC solutions, we believe that it does not directly reflect the performance of the ACOP solutions in supporting admission control.

A good pre-computation mechanism should approximate the supported QoS as precisely as possible. In other words, the error in estimation should be minimized. Since any possible delay constraint is considered, this "error" is *not* a single cost deviation, but an *area* on the Cartesian plane. To illustrate, consider that there are three paths connecting a source to a destination. The QoS parameters of the paths are $(1, 10)$, $(2, 2)$, and $(10, 1)$, where the first element in the tuple reflects the cost of the path while the second element represents the path delay. In this paper, we write the QoS parameter of a path and the constraints of a request as (*cost,delay*). Request (c,d) can be supported by a path with the QoS parameter (c', d'), where $c' \leq c$ and $d' \leq d$. Request $(5, 5)$ is feasible because it can be supported by the path with the QoS parameter $(2, 2)$. Request $(1, 15)$ is also feasible because it can be supported by the path with the QoS parameter $(1, 10)$. However, Request $(1, 1)$ is not feasible because no path can support it. If we plot the QoS parameters of the path on the cost-delay plane, any request that can be supported by any of the paths can be easily identified. Refer to Fig. 1(a), the shaded area is the optimal supported QoS, in which any request that falls in the region is feasible. Thus, a good pre-computation scheme should approximate this area as precisely as possible. The "error" in approximation is the difference in terms of the area between the region of the optimal supported QoS and that of the approximated supported QoS.

While cost deviation is related to the difference in area, it is not sufficient. For example, the shaded areas in Fig. 1(a) and (b) represent the optimal and approximate supported QoS regions, respectively. According to Fig. 1(b), when delay is two, the approximate best cost is three. The cost deviation is $3 - 2 = 1$, as the optimal cost is two. The area of $\{[2, 3] \times [2, 10]\}$ is the "error" in estimating the supported QoS. Any request with the QoS requirements falling in this area is considered as infeasible but actually they are supported by the network. For example, Request $(2, 5)$ is in fact feasible but can be rejected by any approximation algorithms based on the approximate QoS in Fig. 1(b). On the other hand, in Fig. 1(d), the cost deviation with the delay constraint of two is also one. However, we can observe that the "error" in Fig. 1(d) is much smaller than that in Fig. 1(b). Request $(2, 5)$ would be

correctly classified as feasible. Thus, the approximate supported QoS in Fig. 1(d) provides a better network QoS providence than that in Fig. 1(b). The above example illustrates that cost deviation cannot sufficiently reflect the admission control ability of the algorithm, while the "area" does. In this work, we propose a new metric, known as *distortion area*, which is defined as the difference between the approximate supported QoS region and the optimal supported QoS region, to evaluate the accuracy performance of the algorithm for estimating the supported QoS. We first analyze the distortion area of the representative algorithms described in Orda and Sprintson (2003) and Xue et al. (2007). Then, we illustrate how to improve the algorithm to reduce the error.

2. Related work

The MCP problem and RSP (DCLC) problem have been studied extensively (Garroppo et al., 2010). The work in Hassin (1992) focuses on the RSP problem and presents two polynomial algorithms. The author first presented an ε-approximation algorithm by using the basic technique of rounding and scaling with the time complexity $O(\log \log(\frac{UB}{LB})(|E|(n/\varepsilon) + \log \log(\frac{UB}{LB})))$, where UB and LB are the cost metrics of the minimum delay path and the minimum cost path, respectively, $|E|$ is the number of links, and n is the number of nodes. Then, the author applied the basic technique of *interval partitioning* in Sahni (1977) to design a second ε-approximation algorithm with the time complexity of $O(|E|(n^2/\varepsilon) \log(n/\varepsilon))$. Since the time complexity of the first algorithm depends on the upper bound of delay metric of each link, this algorithm is classified in Orda and Sprintson (2003) as a *pseudo-polynomial algorithm*. To the best of our knowledge, the algorithm in Xue et al. (2007) is currently fastest one for precomputing the supported QoS with two additive constraints, which has the complexity of $O(|E|n(\log \log n + 1/\varepsilon))$, where ε is the small positive constant. The smaller ε, the higher accuracy but the higher computational overhead. Chen et al. (2008) design heuristic techniques to improve the performance computational overhead of the algorithm in Goel et al. (2001). As we know, heuristic algorithm cannot provide performance guarantees and the introduced approximation error is difficult to be bounded.

Xue et al. (2008) study the decision version of the problem. Given a connection request with two additive constraints, the proposed algorithm either finds a feasible solution or confirms that there does not exist a source-destination path whose first metric is bounded by the first constraint and whose second weight is bounded by $(1 - \varepsilon)$ times the second constraint. Afterwards, Huang et al. (2012) enhance the algorithm proposed by Xue et al. (2008) by reducing the computational complexity. Li and Zhang (2010) study MCP problem in smart grid and propose a simple heuristic algorithm. Avallone and Ventre (2012) develop a routing algorithm which searches for a feasible path for a given flow request with multiple additive constraints that requires the least number of nodes and links to be turned on. Lu and Zhu (2013)
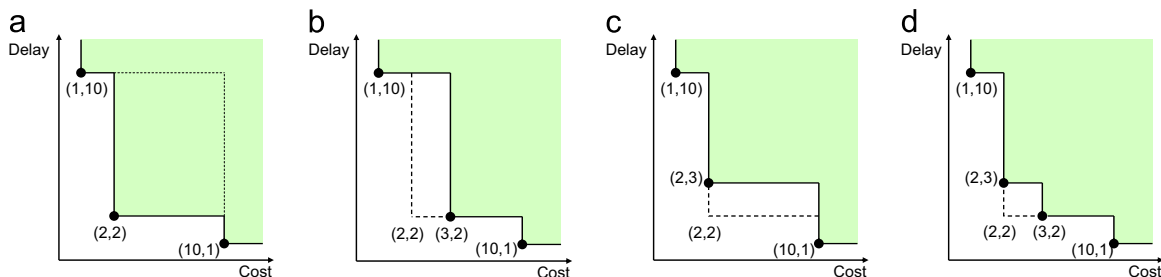

Fig. 1. An illustration for the different approximate solutions. (a) The optimal solution, (b) with cost quantization, (c) with delay quantization, (d) cost and delay quantization.