Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computer Communications 29 (2006) 392–401

computer communications

www.elsevier.com/locate/comcom

Finding a least hop(s) path subject to multiple additive constraints

Gang Cheng, Nirwan Ansari*

Authors are with the Advanced Networking Laboratory, ECE Department, NJIT, Newark, NJ 07012, USA

Received 1 November 2004; revised 9 May 2005; accepted 10 May 2005
Available online 13 June 2005

Abstract

In this paper, for the purpose of saving network resources, we first introduce and investigate a new problem referred to as the least hop(s) multiple additively constrained path (LHMACP) selection, which is NP-complete. Then, we propose the k-shortest paths Extended Bellman-Ford (k-EB) algorithm, which is capable of computing All Hops k-shortest Paths (AHKP) between a source and a destination. Through extensive analysis and simulations, we show that the heuristic algorithm, based on k-EB, is highly effective in finding a least hop path subject to multiple additive constraints with very low computational complexity; it achieves near 100% success ratio in finding a feasible path while minimizing its average hop count.

Keywords: Multiple additively constrained QoS routing; Cost function; NP-complete; Least hop(s)

1. Introduction

The tremendous growth of the global Internet has given rise to a variety of applications that require quality-of-service (QoS) beyond what is provided by the current best-effort IP packet delivery service. One of the challenging issues is to select feasible paths that satisfy different quality-of-service (QoS) requirements. This problem is known as QoS routing. QoS requirements are diverse, subject to demands of different applications. Bandwidth, delay, delay jitter, and loss ratio are the commonly required QoS metrics. These requirements can be classified into three types [1]: concave, additive, and multiplicative. Since the concave type can be easily pruned by selecting the bottleneck, and the multiplicative type can be converted into the additive constraints by the logarithmic operation, the constraints considered in this paper are additive, unless otherwise mentioned. In general, state distribution and routing strategy are the two issues related to QoS routing. State distribution addresses the issue of exchanging the state information throughout the network [2]. Routing strategy is used to find a feasible path meeting the QoS requirements. According to how the state information is aintained and how the search is carried out, routing strategy can be further ivided into three categories [1]: source routing, distributed routing, and hierarchical routing. In this paper, we focus on source routing, and assume that accurate network state information is available to each node. A number of research works have also addressed inaccurate information [3–6], which is, however, beyond the scope of this paper.

Since multiple additively constrained QoS routing has been proved to be NP-complete [7], tackling this problem requires heuristics. The limited path heuristic proposed by Yuan [8] maintains a limited number of candidate paths, say x, at each hop. The computational complexity is $O(x2\ \mathrm{nm})$ for the Extended Bellman-Ford algorithm for two constraints, where m and n are the number of links and nodes, respectively. For the purpose of improving the response time and reducing the computation load on a network, precomputation-based methods [9] have been proposed. Korkmaz and Krunz [10] provided a heuristic with the computational complexity compatible to that of the Dijkstra algorithm in finding the least cost path subject to multiple constraints. An algorithm [11], called A*Prune, is capable of locating multiple shortest feasible paths from the maintained heap in which all candidate paths are stored. The computational complexity of A*Prune is $O(Qn(M+h+\log \mathrm{Q}))$, where Q, M, and h are the number of paths in the heap, the number of constraints, and the maximum number of hops of computed paths, respectively. For the case

* Corresponding author. Tel./fax: +1 973 596 3670.
E-mail address: ansari@njit.edu (N. Ansari).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.05.009

that only inaccurate link state information is available to nodes, approximate solutions [12] have been provided for the Most Probable Bandwidth Delay Constrained Path (MP-BDCP) selection problem by decomposing it into two sub-problems: the Most Probable Delay Constrained Path (MP-DCP) and the Most Probable Bandwidth Constrained Path (MP-BCP). In [13], a heuristic algorithm was proposed based on a linear cost function for two additive constraints; this is an *MCP* (*Multiple Constrained Path Selection*) problem with two additive constraints. A binary search strategy for finding the appropriate value of β in the linear cost function $w_1(p)+\beta w_2(p)$ or $\beta w_1(p)+w_2(p)$, where $w_i(p)$ $(i=1,2)$ are the two respective weights of the path p, was proposed, and a hierarchical Dijkstra algorithm was introduced to find the path. It was shown that the worst-case complexity of the algorithm is $O(\log B(m+n \log n))$, where B is the upper bound of the parameter β. The authors in [14] simplified the multiple constrained QoS routing problem into the shortest path selection problem, in which the Weighted Fair Queuing (*WFQ*) service discipline is assumed. Hence, this routing algorithm cannot be applied to networks where other service disciplines are employed. Similar to [13], LAgrange Relaxation based Aggregated Cost (*LARAC*) was proposed in [15] for the Delay Constrained Least Cost path problem (*DCLC*). This algorithm is based on a linear cost function $c_\lambda=c+\lambda d$, where c denotes the cost, d the delay, and λ an adjustable parameter. It differs from [13] on how λ is defined: λ is computed by Lagrange Relaxation instead of the binary search. It was shown that the computational complexity of this algorithm is $O(m^2\log^4 m)$. However, in [16], for the same problem (*DCLC*), a non-linear cost function was proposed. Many researchers have posed the QoS routing problem as the k-shortest path problem [17,18]. The authors in [19] proposed an algorithm, called TAMCRA, for MCP by using a non-linear cost function and a k-shortest path algorithm. The computational complexity of TAMCRA is $O(kn \log (kn)+k^3mM)$, where k is the number of shortest paths. To solve the delay-cost-constrained routing problem, Chen and Nahrstedt [20] proposed an algorithm, which maps each constraint from a positive real number to a positive integer. By doing so, the mapping offers a 'coarser resolution' of the original problem, and the positive integer is used as an index in the algorithm. The computational complexity is reduced to pseudo-polynomial time, and the performance of the algorithm can be improved by adjusting a parameter, but with a larger overhead. It was shown that only in specially constructed graphs with link weights carefully chosen, NP-complete behavior of QoS routing emerges [21]. Hence, the authors believed that the worst-case behavior (NP-complete) is very unlikely to occur in practical networks and QoS routing is feasible in practice.

Many (ε-approximation algorithms (the solution has a cost within a factor of $(1+\varepsilon)$ of the optimal one) subject to DCLC have been proposed in the literature. Lorenz and Orda [22] presented several ε-approximation solutions for both the DCLC and the multicast tree, in which the one subject to DCLC possesses the best-known computational complexity of $O(nm \log n(\log n)+mn/\varepsilon)$. Hassin [23] presented two ε-approximations algorithms for the Restricted Shortest Path problem (RSP) with complexities of $O((mn/\varepsilon)\log \log U$ and $O(|E|n^2\log(n/\varepsilon)/\varepsilon)$, respectively, where U is the upper bound of the cost of the path computed.

It can be observed from the above review that existing solutions suffer either high computational complexities or low success ratio in finding a feasible path. In this paper, based on a novel solution to All Hops k-shortest Paths selection (AHKP), we propose a high performance routing algorithm for finding the Least Hop(s) Multiple Additive Constrained Path (LHMACP). By extensive simulations, we show that our proposed algorithm not only achieves near 100% success ratio in finding a feasible path, but also essentially minimizes the average number of hops of the computed feasible paths. As a result, network resources can be saved with our proposed routing algorithm.

The rest of the paper is organized as follows. The problem is formulated in Section 2. The k-shortest paths Extended Bellman-Ford (k-EB) algorithm, which is capable of computing all hops k-shortest paths between a source and a destination, is proposed in Section 3. Based on k-EB, we present a high performance heuristic algorithm which achieves a high success ratio in finding the least hop(s) multiple additively constrained paths in Section 4. In Section 5, simulation results are presented. Finally, concluding remarks are given in Section 6.

2. Problem formulation

Since we only consider additive constraints in this paper, without loss of generality, the problem is formulated as follows:

Definition 1. *Least Hop*(*s*) *Multiple Additively Constrained Path Selection* (*LHMACP*): Assume a network is modeled as a directed graph $G(N,E)$, where N is the set of all nodes and E is the set of all links. Each link connected from node u to v, denoted by $e_{u,v}=(u,v)\in E$, is associated with M additive parameters: $w_i(u,v)\geq 0$, $i=1,2,\ldots,M$. Given a set of constraints $(c_1,c_2,\ldots,c_M)$ and a pair of nodes s and t, LHMACP is to find a least hop(s) path p from s to t subject to $W_i(p)=\sum_{e_{u,v}\in p} w_i(u,v)<c_i$, for all $i=1,2,\ldots,M$.

Definition 2. Any path p from s to t that meets the requirement, $W_i(p)=\sum_{e_{u,v}\in p} w_i(u,v)<c_i$, for all $i=1,2,\ldots,M$, is a feasible path.

We denote p_1+p_2 as the concatenation of two paths p_1 and p_2, and $c(p)$ as the cost of path p. Note that, given two paths, p_1 and p_2, and their costs, if the cost of a path p is defined as $c(p)=f(W_1(p), W_2(p),\ldots,W_m(p))$, where $f(\cdot)$ is a cost function, the computational complexity of computing the cost of p_1+p_2, $c(p_1+p_2)=f(W_1(p_1)+W_1(p_2),$