Finding All Hops Shortest Paths

Gang Cheng and Nirwan Ansari, *Senior Member, IEEE*

***Abstract*—In this letter, we introduce and investigate a new problem referred to as the All Hops Shortest Paths (AHSP) problem. The AHSP problem involves selecting, for all hop counts, the shortest paths from a given source to any other node in a network. We derive a tight lower bound on the worst-case computational complexities of the optimal comparison-based solutions to AHSP.**

***Index Terms*—Quality of service (QoS), routing.**

I. Introduction

ONE OF THE challenging issues for high-speed packet switching networks to accommodate various applications with different quality-of-service (QoS) requirements is to select feasible paths. This problem is known as QoS routing. However, it has been proved that multiple additively constrained QoS routing is NP-complete [1]. Many proposed source routing algorithms tackle this problem by transforming it into the shortest path selection problem, which is P-complete, with an integrated cost function that maps the multi-constraints of each link into a single cost. Given a set of constraints $(\alpha_1, \alpha_2, \ldots, \alpha_M)$ and a network that 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, assume each link connected from node u to v, denoted by $e_{u,v} = (u, v) \in E$, is associated with M randomly distributed additive parameters: $w_i(u, v) \geq 0, i = 1, 2, \ldots, M$, and define $P_r\{W_1(p) \leq \alpha_1, W_2(p) \leq \alpha_2, \ldots, W_M(p) \leq \alpha_M \,|\, C(p) = u, H(p) = n\}$ as the probability that a path p is a feasible path with $C(p) = u$, and its hop count, $H(p) = n$, where $C(p)$ is the cost of p, which is a function of the weights of the links on p, and $W_i(p) = \sum_{e_{u,v} \in p} w_i(u, v)$. The probability of the shortest path to be a feasible path may not be the largest in all paths. Therefore, computing a feasible path among all hops shortest paths, instead of only the shortest path, can increase the success ratio of finding a feasible path. In this letter, we introduce and investigate a new problem referred to as All Hops Shortest Paths (AHSP) problem, defined below.

Definition 1—All Hops Shortest Paths (AHSP) Problem: 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 an additive weight $c(u, v)$. Given a source node $s \in N$ and maximal hop count $H, H < n$, find, for each hop count value $h, 1 \leq h \leq H$, and any other destination node $u \in N$, the shortest (i.e., the least weight) h-hop path from s to u if such a path does exist. In this letter, we will refer to the length (i.e., cost) of a path as the sum of its link weights.

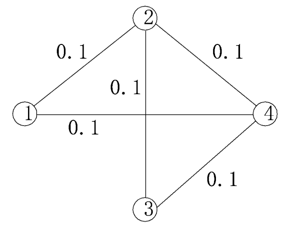

Fig. 1. A 4-nodes network.

Note that AHSP is different from add-AHOP [2]. For each hop count value $h, 1 \leq h \leq H$, add-AHOP is to select the shortest path from a given source to a destination that has a hop count no larger than h; while AHSP is to select the shortest path from the source to the destination that has a hop count equal to h if such a path physically exists (it might be more appropriate to call AHOP as All-Hops-Constrained Optimal Path problem instead of All Hops Optimal Path problem). For example, as shown in Fig. 1, the cost of each link in the 4-node network is 0.1. Let $(\alpha_1, \alpha_2, \ldots, \alpha_h)$ represents an h-hop path from node α_1 to node α_h sequentially traversing nodes $\alpha_1, \alpha_2, \ldots, \alpha_h$, respectively. Given $H = 3$, since (1, 4) is the shortest path from node 1 to node 4 with a hop count of 1, for each hop count $h \in \{1, 2, 3\}$, the shortest path selected by add-AHOP with a hop count no larger than h from node 1 to node 4 would always be $(1, 4)$; while the h-hop shortest paths selected by AHSP from node 1 to node 4 are (1, 4), (1, 2, 4), and (1, 2, 3, 4), respectively, when h equals to 1, 2, and 3. Hence, solving add-AHOP does not need to solve AHSP. On the other hand, if AHSP is solved, add-AHOP is also solved.

AHSP seems to be more difficult than add-AHOP because in addition to the paths selected by add-AHOP, AHSP involves selecting paths that are not selected by add-AHOP, i.e., since the shortest path from the source to the destination that has a hop count no larger than h must be a $\tilde{h}$-hop shortest path from the source to the destination, where $\tilde{h}$ is its hop count and $\tilde{h} \leq h$; the paths selected by AHSP must include the paths selected by add-AHOP. However, in this letter, we prove that the comparison-based optimal solutions for both add-AHOP and AHSP have the same order of worst-case computational complexities, where the optimal solutions are referred to as the solutions possessing the minimum worst-case computational complexity. Therefore, since AHSP yields more paths than AHOP, given a QoS routing algorithm in which a solution to AHOP, e.g., Bellman-Ford algorithm, is deployed, we can increase the success ratio in finding a feasible path by replacing the solution

Manuscript received June 4, 2003. The associate editor coordinating the review of this letter and approving it for publication was Prof. M. Ma. This work has been supported in part by the New Jersey Commission on Higher Education via the NJI-TOWER project, and the New Jersey Commission on Science and Technology via NJWINS.

The authors are with the Advanced Networking Laboratory, Electrical and Computer Engineering Department, Newark Institute of Technology, Newark, NJ 07012 USA (e-mail: ansari@njit.edu).

Digital Object Identifier 10.1109/LCOMM.2004.823365

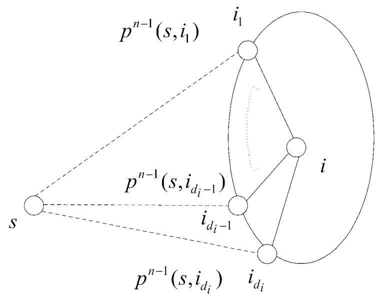

Fig. 2. Illustration of $p^n(s,i)$.

to AHOP with an optimal solution to AHSP without increasing the computational complexity.

II. A Lower Bound on the Worst-Case Computational Complexities of the Comparison-Based Solutions

Before we proceed to further analysis, we first present the following definition [3].

Definition 2: A path-comparison-based shortest-path algorithm Γ, which accepts as input a graph G and a weight function, can perform all standard operations, but the only way it can access the edge weights is to compare the weights of two different paths [3].

Denote d_i as the degree of node i and $i_1, i_2, \ldots, i_{d_i}$ as its neighboring nodes. If there does not exist a physical link from the source node s to any other node i, assume there exists a virtual link $\hat{e}(s,i)$ from s to i with a cost of infinity. Denote $p^n(s,i)$ as an n-hop path from the source s to i (see Fig. 2). Then,

1) $\forall i \in N, p^1(s,i) = e(s,i)$ (if, in reality, no link between the source s and node i exists, $p^1(s,i) = \hat{e}(s,i)$).
2) $p^n(s,i)$ represents the shortest[1] n-hop path among the paths $p^{n-1}(s,i_d) + e(i_d,i), 1 \le d \le d_i$. If, in reality, no n-hop path from s to i exists, we assume that there exists a virtual n-hop path whose weight is infinity.

Denote D_i^n as the weight of $p^n(s,i)$; $\pi^n(i)$ represents the predecessor node of i along the path.

Theorem 1: $p^n(s,i), 1 \le n \le H$, are the solutions to AHSP.

Proof: When $n = 1$, from the definition of the initial value of $D_i^1 (i \neq s), p^1(s,i)$ is the one hop shortest path from s to $i, i \in \{1, 2, \ldots, N\}$.

We assume that the proposition is correct for $n = k$. We want to prove by deduction that it is true for $n = k+1$.

Assume when $n = k+1, \exists j \neq s$ such that $p^{k+1}(s,j)$ is not the shortest path in all $(k+1)$-hop paths from s to j (D_j^{k+1} is larger than the cost of the $(k+1)$-hop shortest path from s to j). Further assume path $\hat{p}^{k+1}(s,j)$ is the shortest path in all $(k+1)$-hop paths from s to j, the predecessor node of node j in $\hat{p}^{k+1}(s,j)$ is d, the path from s to d in $\hat{p}^{k+1}(s,j)$ is $\hat{p}^k(s,d)$ (note that $\hat{p}^k(s,d)$ may not be the shortest k-hop path from s to d, and by the earlier assumption, since a k-hop path exists, $p^k(s,d)$ is the shortest path from s to d), the cost of $\hat{p}^{k+1}(s,j)$ is c, and the cost of $\hat{p}^k(s,d)$ is c'. Thus

$$c < D_j^{k+1}. \tag{1}$$

Since $\hat{p}^{k+1}(s,j)$ is resulted by concatenating $\hat{p}^k(s,d)$ with link $e(d,j)$, if $\hat{p}^k(s,d) = p^k(s,d)$ or $c' = D_d^k$,

$$c = D_d^k + c(d,j) \ge \min_l \left[D_l^k + c(l,j)\right] = D_j^{k+1} \tag{2}$$

which contradicts (1). Hence, $\hat{p}^k(s,d)$ is not the k-hop shortest path from s to d, i.e.,

$$\hat{p}^k(s,d) \neq p^k(s,d) \tag{3}$$

and

$$c' > D_d^k. \tag{4}$$

So, the cost of $\hat{p}^{k+1}(s,j)$ is

$$\begin{aligned} c &= c' + c(d,j) > D_d^k + c(d,j) \\ &\ge \min_l \left[D_l^k + c(l,j)\right] = D_j^{k+1} \end{aligned} \tag{5}$$

which contradicts (1), implying that $\hat{p}^{k+1}(s,j)$ is not the $(k+1)$-hop shortest path from s to j, contradicting to assumption. So, when $n = k+1, p^{k+1}(s,i), i \in \{1, 2, \ldots, N\}$, is the shortest path among all $(k+1)$-hop path from s to i.

Thus, for any node $i \in \{1, 2, \ldots, N\}$, if at least one n-hop path from s to i physically exists, the path $p^n(s,i)$ must be the shortest path among all n-hop paths from s to i, i.e., $p^n(s,i), 1 \le n \le H$, are the solutions to AHSP. ■

By Theorem 1, we know that the n-hop shortest path from s to i is the shortest path of the paths that are constructed by concatenating the $(n-1)$-hop shortest paths from s to the neighboring nodes of $i$with the corresponding links. Next, we provide a tight lower bound on the worst-case computational complexity of the optimal comparison-based solution to AHSP based on Theorem 1.

Theorem 2: The optimal comparison-based solution to AHSP has the worst-case computational complexity of $O(H|E|)$.

Proof: First of all, we prove that the upper bound on the worst-case computational complexities of the optimal comparison-based solutions to AHSP is $O(H|E|)$. Note that the $(h+1)$-hop least weight path from s to i is the least weight path among the paths constructed by concatenating the h-hop least weight paths from s to the neighboring nodes of i with the corresponding links. Hence, in order to compute the $(h+1)$-hop shortest path from s to i, the h-hop least weight paths from s to the neighboring nodes of i should be computed first. Moreover, given a node i, the $(h+1)$-hop least weight path from s to i can be achieved with the computational complexity of $O(d_i)$ by the definition of the comparison-based algorithm and Theorem 1. Hence, given the h-hop shortest paths from s to all other nodes, the $(h+1)$-hop shortest paths from s to all other nodes can be computed with the computational complexity of $O(\sum_{i \in N, i \neq s} d_i) = O(|E|)$. Define f_H as the computational complexity bound on the optimal solutions (i.e., the ones with

[1] Since the cost of a path is the length of the path, the least cost n-hop path is the shortest n-hop path.

the minimum worst-case computational complexity) to AHSP, where H is the maximum hop. Therefore

$$\begin{aligned} f_H &\leq f_{H-1} + O(|E|) \\ &\Rightarrow f_H - f_{H-1} \leq O(|E|) \\ &\Rightarrow f_H - f_1 \leq O((H-1)|E|) \\ &\Rightarrow f_H \leq O(H|E|) \end{aligned} \tag{6}$$

implying that the optimal comparison-based solution to AHSP has a computational complexity no larger than $O(H|E|)$.

On the other hand, since AHSP computes all the paths AHOP computes, the computational complexity of the optimal comparison-based solution to AHSP must be no less than that of the optimal comparison-based solution to add-AHOP. It has been proved in [2] that the tight lower bound on the computational complexity of the optimal comparison-based solution to add-AHOP is $O(|V|^3)$, where $|V|$ is the number of nodes. Therefore,

$$O(|V|^3) \leq f_H \leq O(H|E|). \tag{7}$$

However, only when $|E| = O(|V|^2)$ and $H = O(|V|)$, $O(|V|^3) = O(|V||E|)$. Hence, the optimal comparison-based solution to AHSP has the worst-case computational complexity of $O(H|E|)$. ■

By Theorem 2, we know that the worst-case computational complexity of an optimal comparison-based solution to AHSP is the same as that of an optimal comparison-based solution to add-AHOP. Note that link weights are assumed to be additive in AHSP in this letter. Hence, Theorem 2 may not be applicable to the case that they are not additive, e.g., concave and multiplicative (it may not be appropriate to call the problem as AHSP anymore). However, for the case when the link weights are concave, it can be proved that the n-hop path with the largest weight from the source to node i can be computed by letting $p^n(s,i)$ represent the n-hop path with the largest weight among the paths $p^{n-1}(s,i_d) + e(i_d,i), 1 \leq d \leq d_i$ (when the link weights are concave, it has become the problem of finding all hops paths with the largest weights). Moreover, multiplicative link weights can be converted to additive weights by using the logarithm function. Therefore, similar to Theorem 2, we still can prove that $O(H|E|)$ is an upper bound on the worst-case computational complexity of the optimal comparison-based solutions.

III. Conclusion

In this letter, we have introduced and investigated a new problem referred to as the All Hops Shortest Paths (AHSP) problem. A tight lower bound on the worst-case computational complexity of the optimal comparison-based solution to AHSP has also been derived.

References

[1] Z. Wang and J. Crowcroft, "Quality of service routing for supporting multimedia applications," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1228–1234, 1996.

[2] R. Guerin and A. Orda, "Computing shortest paths for any number of hops," *IEEE/ACM Trans. Networking*, vol. 10, pp. 613–620, 2002.

[3] D. R. Karger, D. Koller, and S. J. Phillips, "Finding the hidden path: Time bounds for all-pairs shortest paths," *SIAM J. Computing*, vol. 22, pp. 1199–1217, 1993.