

On Algorithm for All-Pairs Most Reliable Quickest Paths*

Young-Cheol Bang¹, Inki Hong¹, and Hyunseung Choo^{2,**}

¹ Department of Computer Engineering, Korea Polytechnic University
Kyunggi-Do, Korea

{ybang, isslhong}@kpu.ac.kr

² School of Information and Communications Engineering, Sungkyunkwan University
Suwon, Kyunggi-Do, Korea
choo@ece.skku.ac.kr

Abstract. The reliability problem of the quickest path deals with the transmission of a message of size σ from a source to a destination with both the minimum end-to-end delay and the reliability of the path over a network with bandwidth, delay, and probability of fault free on the links. For any value of message size σ , we present algorithm to compute all-pairs most-reliable quickest paths each with time complexity $O(n^2m)$, where n and m are the number of nodes and the number of arcs or links in the network, respectively.

1 Introduction

We consider point-to-point computer networks represented by a graph $G = (V, A)$ with n nodes and m arcs or links. Any node can be reached from any other node in this network, and two nodes are connected by at most single directed link in each direction. Each directed link $l = (i, j) \in A$ has a bandwidth $B(l) \geq 0$, delay $D(l) \geq 0$, and probability of fault free $0 \leq \pi(l) \leq 1$. A message of σ units can be sent along the link l in $T(l)\sigma / B(l) + D(l)$ time with reliability $\pi(l)$ as in [4].

Consider a simple path P from i_0 to i_k given by $(i_0, i_1), (i_1, i_2), \dots, (i_{k-1}, i_k)$, where $(i_j, i_{j+1}) \in A$, for $j = 0, 1, 2, \dots, (k-1)$, and all i_0, i_1, \dots, i_k are distinct. Subsequently, a simple path is referred to simply as a path. The delay of this path P , denoted by $D[P]$,

is given by $\sum_{j=0}^{k-1} D(l_j)$, where $l_j = (i_j, i_{j+1})$. The bandwidth of this path is defined as

$B(P) = \min_{j=0}^{k-1} B(l_j)$. The reliability of P , denoted by $R(P)$, is $\prod_{j=0}^{k-1} \pi(i_j, i_{j+1})$. The end-

to-end delay of the path P in transmitting a message of size σ is $T(P) = \sigma / B(P) + D[P]$ with reliability $R(P)$. Let $b_1 < b_2 < \dots < b_r$ and G_b denote the distinct values of

* This paper was supported in part by Brain Korea 21 and University ITRC project.

** Dr. Choo is the corresponding author.

$B(l)$, $l \in E$ and the sub-network with all edges of G whose bandwidth is greater than or equal to b , respectively.

The path P from s to d is the most-reliable (MR) if $R(P)$ is the maximum among all paths from s to d . The path P is the quickest for message size σ if $T(P)$ is the minimum among all paths from s to d . The path P is the quickest most-reliable (QMR) if it is the quickest for σ among all MR paths from s to d . The P is the most-reliable quickest (MRQ) if it has highest reliability among all quickest paths from s to d for σ .

In this paper, we consider the all-pairs versions of computing MRQ paths with respect to any size of σ . The all-pairs version of the classical quickest path problem was solved in [2, 3] with time complexity of $O(n^2m)$. By applying the algorithms of Xue [4] for each $s \in V$, we can compute MRQ paths between all pairs s and d with time complexity $O(nrm + rn^2 \log n)$; since $r \leq m$, we have the complexity $O(nm^2 + n^2m \log n)$. In this paper, we present $O(n^2m)$ time algorithm to compute all-pairs MRQ paths, which match the best-known complexity for the all-pairs classical quickest path problem.

The rest of paper is organized as follows. In section 2, we present details of algorithm to compute the MRQ paths for all pairs of nodes in a given network. Section 3 summarizes our researches.

2 All-Pairs Most-Reliable Quickest Paths for Any Size of Message

To compute an MRQ path from s to d , we have to “account” for all quickest paths from s to d . Note that all-pairs quickest path algorithm (AQP) [3] returns a quickest path from s to d , which may not be a MRQ path, and hence a simple condition similar to line 8 of all-pairs QMR algorithm (AQMR) [1] does not work. In particular, it is not sufficient to check if an edge l is on a quickest path P_1 with bandwidth $B(l)$; in fact, l can be on a quickest path with any $b = B(P_1) \leq B(l)$. In our algorithm, we compute the largest of such b and place l at an appropriate step in the computation, which is an iterative process similar to AQMR. Let $t[u, v]$ represent the end-to-end delay of quickest path from u to v for σ .

To compute MRQ paths, we first compute all-pairs quickest paths in G using AQP with the following enhancement. For each bandwidth value b_k and pair $u, v \in V$, we store a matrix $[d_{b_k}[u, v]]$ where $d_{b_k}[u, v]$ is the delay of the shortest path from u to v in G_{b_k} . These matrices can be computed for $b = b_r, b_{r-1}, \dots, b_1$ during the execution of AQP. For each (σ_i, σ_{i+1}) , $1 \leq i \leq r - 1$, we define

$\Theta(\sigma_i, \sigma_{i+1}, u, v) = \{b_k \mid B(P_k) \text{ such that } P_k \text{ is the quickest path for any } \sigma \in (\sigma_i, \sigma_{i+1})\}$ if such b_k exists, and \emptyset otherwise. In this case, if $\sigma_i \neq \sigma_{i+1}$ then $\sigma \in (\sigma_i, \sigma_{i+1})$ represents $\sigma_i < \sigma < \sigma_{i+1}$, otherwise (σ_i, σ_{i+1}) equals to the intersection point $\sigma_{i,i+1}$.

Lemma 1. (i) $\Theta(\sigma_i, \sigma_{i+1}, u, v) \neq \emptyset$ if and only if there is a shortest path from u to v in G_b for some $b \in \Theta(\sigma_i, \sigma_{i+1}, u, v)$.

(ii) \exists the quickest path from u to v for $b \in \Theta(\sigma_i, \sigma_{i+1}, u, v)$ if and only if $\exists b_k$ such that $b_k \in \Theta(\sigma_i, \sigma_{i+1}, u, v)$

Lemma 2. All $\Theta(\sigma_i, \sigma_{i+1}, u, v)$ can be computed with the time complexity of $O(n^2m)$ for all $u, v \in E$.

In AMRQ, we organize the sets $\Theta(\sigma_i, \sigma_{i+1}, u, v)$'s as stacks with bandwidths decreasing top to bottom. Let $\Omega[u, v]$ denote the queue to store Θ s in order of which each Θ is computed. We use AQP [3] to compute $[d_b[u, v]]$ in line 1. In line 2-4, we compute all $\Theta(\sigma_i, \sigma_{i+1}, u, v)$'s for all pairs $u, v \in V$ with time complexity $O(n^2m)$. There are $O(n^2m)$ iterations in the rest of the algorithm, where links are considered in non-increasing order of bandwidth with which they participate in quickest paths (if at all). In each iteration, we consider the current link bandwidth $B(l)$, and pair $u, v \in V$. Lines 12-17 compute the maximum bandwidth with which the link l is used in a quickest path from u to v . The reliability of new path via l from u to v is then computed and the existing value and MRQ path are replaced appropriately in lines 18-20. Consider that as a result of while loop in lines 12-17, the retrieved bandwidth $b_{[u, v]}$ is strictly smaller than $B(l)$ if $b_{[u, v]}$ corresponds to link l_1 , no more pop operations on $\Theta(\sigma_i, \sigma_{i+1}, u, v)$ will be performed until all links with bandwidths in the range $[B(l_1), B(l)]$ have been retrieved from the heap and processed. For each pair $u, v \in V$, this algorithm can be viewed in terms of alternating subsequences of top operations on arc_heap, dequeue operations on queue $\Omega[u, v]$, and pop operations on stack $\Theta(\sigma_i, \sigma_{i+1}, u, v)$ with no backtracking involved. In actual execution, however, all these subsequences corresponding to various $u - v$ pairs are intermingled among themselves as well as subsequences of top operations.

Algorithm AMRQ (G, D, B, π)

/* MRQP(u, v, σ_i, σ_j) maintains MRQ path from u to v for (σ_i, σ_j) */

/* $P[u, v]$ maintains a currently selected MRQ path from u to v */

1. compute $[d_b[u, v]]$ using AQP (G, B, D)
2. **for** each pair $u, v \in V$ **do**
3. **for** each interval of σ **do**
4. compute stack $\Theta(u, v, \sigma_i, \sigma_j)$ and store to $\Omega[u, v]$ with $i \leq j$
5. **for** each pair $u, v \in V$ **do**
6. $\Theta(u, v, \sigma_i, \sigma_j) = \text{dequeue}(\Omega[u, v])$
7. $b_{[u, v]} = \text{pop}(\Theta(u, v, \sigma_i, \sigma_j))$;
8. arc_heap = top_heavy heap of all edges of G according to the bandwidth

```

9.  while not arc_heap  $\neq \emptyset$  do
10.    $(i, j) = \text{top}(\text{arc\_heap});$  let  $l = (i, j);$ 
11.   for each pair  $u, v \in V$  do
12.     while  $(B(l) < b_{[u,v]})$  do
13.       if  $(\Theta(u, v, \sigma_i, \sigma_j) \neq \emptyset)$  then
14.          $b_{[u,v]} = \text{pop}(\Theta(u, v, \sigma_i, \sigma_j));$ 
15.       else
16.          $\Theta(u, v, \sigma_i, \sigma_j) = \text{dequeue}(\Omega[u, v])$  if  $\Omega[u, v] \neq \emptyset$ 
17.          $b_{[u,v]} = \text{pop}(\Theta(u, v, \sigma_i, \sigma_j));$ 
18.       if  $(B(l) \geq b_{[u,v]})$  and
19.          $(d_{b_{[u,v]}}[u, v] = d_{b_{[u,v]}}[u, i] + D(i, j) + d_{b_{[u,v]}}[j, v])$  then
20.            $\Phi[u, v] \leftarrow \min\{ \Phi[u, v], \Phi[u, i] + \pi^i(i, j) + \Phi[j, v] \};$ 
           /* update routing table for MRQ path from u to v for  $[\sigma_i, \sigma_j]$  */
            $\text{MRQ\_RT}(u, v, \sigma_i, \sigma_j) = P[u, i] + l(i, j) + P[j, v]$  if any

```

Theorem 1. The all-pairs most reliable quickest paths for any size of message can be computed by algorithm AMRQ with time complexity $O(n^2m)$ and space complexity $O(n^2m)$.

3 Conclusion

We presented algorithms to compute most-reliable quickest and quickest most-reliable paths between all pairs of nodes in a network. These algorithms match the best known computational complexity for the classical all-pairs quickest path problem, namely without the reliability considerations.

References

1. Y. C. Bang, H. Choo, and Y. Mun, Reliability Problem on All Pairs Quickest Paths, ICCS2003, LNCS 2660, pp. 518-523, 2003
2. G. H. Chen and Y. C. Hung, On the quickest path problem, *Information Processing Letters*, vol 46, pp. 125-128, 1993
3. D. T. Lee and E. Papadopoulou, The all-pairs quickest path problem, *Information Processing Letters*, vol. 45, pp. 261-267, 1993
4. G. Xue, End-to-end data paths: Quickest or most reliable?, *IEEE Communications Letters*, vol. 2, no. 6, pp. 156-158, 1998