A Distributed Algorithm For Finding All Best Swap Edges Of A Minimum Diameter Spanning Tree

Sai Srishti[1], Divya Sharma[2], Kaushal Kishor[3]
[1,2,3]Department of Information Technology, ABESIT, Ghaziabad, Uttar Pradesh, India

Abstract- The rapid development of computer network system brings both a great convenience and new security threats for users. Network security problem generally includes network system security and data security. Specifically, it refers to the reliability of network system, confidentiality, integrity and availability of data information in the system. Network security problem exists through all the layers of the computer network, and the network security objective is to maintain the confidentiality, authenticity, integrity, dependability, availability and audit-ability of the network. This paper introduces the network security technologies mainly in detail, including authentication, data encryption technology, firewall technology, intrusion detection system (IDS), antivirus technology and virtual private network (VPN). Network security problem is related to every network user, so we should put a high value upon network security, try to prevent hostile attacks and ensure the network security.
Keywords – VPN, IDS, Routing, MDST, Telecommunication

I. INTRODUCTION

Communication in networks suffers, if a link fails. When the link are edges of a tree that has been chosen from an underlying graph of all possible links, a broken link even disconnects the network. Most often, the link is restored rapidly. A good policy to deal with this sort of transient link failures is swap rerouting, where the temporarily broken link is replaced by a single swap link from the underlying graph. A rapid replacement of a broken link by a swap link is only possible if all swap links have been precomputed. The selection of high quality swap links is essential; it must follow the same objective as the originally chosen communication subnetwork.

We are interested in a minimum diameter tree in a graph with edge weights (so as to minimize the maximum travel time of messages). Hence, each swap link must minimize (among all possible swaps) the diameter the tree that results from swapping. We propose a distributed algorithm that efficiently computes all of these swap links, and we explain how to route messages across swap edges with a compact routing scheme. Finally, we consider the computation of swap edges in an arbitrary spanning tree, where swap edges are chosen to minimize the time required to adapt routing in case of a failure, and give efficient distributed algorithms for two variants of this problem. A distributed algorithm is an algorithm designed to run on computer hardware constructed from interconnected processors. Distributed algorithms are used in many varied application areas of distributed computing, such as telecommunications, scientific computing, distributed information processing, and real-time process control. Standard problems solved by distributed algorithms include leader election, consensus, distributed search, spanning tree generation, mutual exclusion, and resource allocation.

Distributed algorithms are a sub-type of parallel algorithm, typically executed concurrently, with separate parts of the algorithm being run simultaneously on independent processors, and having limited information about what the other parts of the algorithm are doing. One of the major challenges in developing and implementing distributed algorithms is successfully coordinating the behavior of the independent parts of the algorithm in the face of processor failures and unreliable communications links. The choice of an appropriate distributed algorithm to solve a given problem depends on both the characteristics of the problem, and characteristics of the system the algorithm will run on such as the type and probability of processor or link failures, the kind of inter-process communication that can be performed, and the level of timing synchronization between separate processes. For communication in computer networks, often only a subset of the available connections is used to communicate at any given time. If all nodes are connected using the smallest number of links, the subset forms a spanning tree of the network. This has economic benefits compared to using the entire set of available links, assuming that merely keeping a link active for potentially sending messages induces some cost. Furthermore, as only one path exists between any communication pair, a spanning tree simplifies routing and allows small routing tables. Depending on the purpose of the network, there is a variety of desirable properties of a spanning tree. We are interested in a Minimum Diameter Spanning Tree (MDST), i.e., a tree that minimizes the largest distance between any pair of nodes, thus minimizing the worst case length of any transmission path, even if edge lengths are not uniform. The importance of minimizing the diameter of a spanning tree has been widely recognized; essentially, the diameter of a network provides a lower bound (and often even an exact one) on the computation time of most algorithms in which all nodes participate.

II. EXISTING SYTEM:

According to the previous technique, a message follows the normal routing table information unless the next hop has failed; in this case, it is redirected towards a pre computed link, called swap; once this link has been crossed, normal routing is resumed. The choice of the swap edge is done according to some optimization criteria on the resulting new route. The amount of pre computed information required in addition to the routing table is rather small: a single link per each destination. Several efficient serial algorithms have been presented to compute this information for several optimization criteria distance, maximum, sum, increment. Only the algorithm corresponding to distance has been efficiently implemented in a distributed environment, while for the other optimization criteria no distributed solution has been devised yet.

In a network communication system, frequently messages are routed along the Minimum Diameter Spanning Tree (MDST) of the network, to minimize the maximum delay in delivering a message. When a transient edge failure occurs, it is important to choose a temporary replacement edge which minimizes the diameter of the new spanning tree. Such an optimal replacement is called the best swap. [4]

We present a new algorithm, which solves the problem of distributively finding a minimum diameter spanning tree of any (non-negatively) real-weighted graph G = (V;E;w). As an intermediate step, we use a new, fast, linear-time all-pairs shortest paths distributed algorithm to find an absolute center of G. The resulting distributed algorithm is asynchronous, it works for named asynchronous arbitrary networks and achieves O (|V|) time complexity and O(|V||E|) message complexity. [2]

We introduce a new recovery scheme that needs only one extra backup routing table for networks employing shortest-path routing. By pre computing this backup table, the network recovers from any single link failure immediately after the failure occurs. To compute the backup routing table for this scheme, we use an almost linear time algorithm to solve the {r, v}-problem, which is a variation of the best swap problem presented by Nardelli et al. We further show that the same solution can be computed in exactly linear time if the underlying graph is unweighted. [3]

Disadvantage: Once this link has been crossed, normal routing is resumed. The choice of the swap edge is done according to some optimization criteria on the resulting new route.

III. PROPOSED ALGORITHM

For communication in computer networks, often only a subset of the available connection is used to communicate at given time. If all nodes are connected using the smallest number of links, the subset forms a spanning tree of the networks. This has economic benefits compared to using the entire set of available links, assuming that merely keeping a link active for potentially sending messages induces some cost. Furthermore, as only one path exists between any communication pair, a spanning tree simplifies routing and allows small routing tables.

Depending on the purpose of the network, there is a variety of desirable properties of a spanning tree. We are interested in a Minimum Diameter Spanning Tree (MDST), i.e., a tree that minimizes the largest distance between any pair of nodes, thus minimizing the worst case length of any transmission path, even if edge lengths are not uniform. The importance of minimizing the diameter of a spanning tree has been widely recognized, the diameter of a network provides a lower bound (and often even an exact one) on the computation time of most algorithms in which all nodes participate. One downside of using a spanning tree is that a single link failure disconnects the network. Whenever link failures are transient, i.e., a failed link soon becomes operational again, the momentarily best possible way of reconnecting the network is to replace the failed link by a single other link, called a Swap link. Among all possible swap links, one should choose a best swap with respect to the original objective that is in our case, a swap that minimizes the diameter of the resulting swap tree.

Advantage:The distributed computation of all best swaps has the further advantage of gaining efficiency (against computing swap edges individually), because dependencies between the computations for different failing edges can be exploited.

IV. PROBLEM STATEMENT:

A communication network is an undirected graph G = (V, E), with n = |V | vertices and m = |E| edges. Each edge e 2 E has a non-negative real length l(e). The length |P| of a path P = {p1, . . . , pr} is the sum of the lengths of its edges, and the distance d(x, y) between two vertices x, y is the length of a shortest path between x and y. Note that throughout the paper, we measure distances in the given spanning tree T, not in the underlying graph G itself. The hop length ||P|| := r-1 of a path P is the number of edges that P contains. Throughout the paper, we are only dealing with simple paths. Given a spanning tree T = (V,E(T)) of G, let

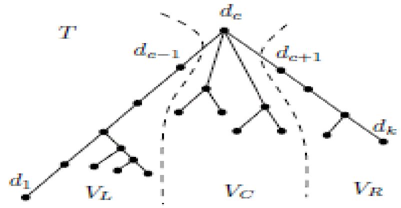

Fig. 1. A minimum diameter spanning tree T.

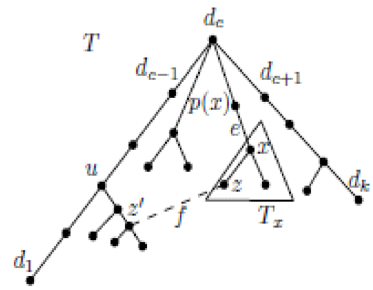

Fig. 2. A Swap Edge f = (z, z0) for e = (x, p(x)).

D (T):= {d1, d2, . . . , dk} denote a diameter of T, that is, a longest path in T (see Fig. 1). Where no confusion arises, we abbreviate D(T) with D. Furthermore, define the center dc of D as a node such that the lengths of DL: = {d1, d2, . . . , dc} and DR: = {dc, dc+1, . . . , do} satisfy |DL| ¸|DR| and have the smallest possible difference |DL| |DR|. The set of neighbors of a node z (excluding z itself) in G and in T is written as NG(z) and NT (z) µ NG(z), respectively.
Let T be rooted at dc, and let, for each node x ≠ dc, node p(x) be the parent of x and C(x) the set of its children. Furthermore, let Tx = (V (Tx),E(Tx)) be the sub tree of T rooted at x, including x. Let VL (L stands for “left”) be the set of nodes in the sub tree rooted at dc-1, VR the set of nodes in the sub tree rooted at dc+1, and VC all other nodes. Now, the removal of any edge e = (x, p(x)) of T partitions the spanning tree into two trees Tx and T\Tx (see Fig. 2), where T\Tx denotes the graph with vertex set V (T)\V (Tx) and edge set E(T)\E(Tx)\{(x, p(x))}. Note that T\Tx does not contain the node x. A swap edge f for e is any edge in E\E (T) that (re-)connects Tx and T\Tx, i.e., for which Te/f: = (V (T), E (T)\ {e} U {f}) is a spanning tree of G\ {e}:= (V, E\{e}).
Let S (e) be the set of swap edges for e. A best swap edge for e is any edge f € S(e) for which |D(Te/f)| is minimum. A local swap edge of node z for some failing edge e is an edge in S (e) adjacent to z. The distributed all best swaps problem for a MDST is the problem of finding for every edge e € E(T) a best swap edge (with respect to the diameter), or to determine that no swap edge for e exists. Throughout the paper, let e = (x, p(x)) denote a failing edge and f = (z, z0) a swap edge, where z is a node inside Tx, and z0 a node in T\Tx

Algorithm used:
F Algorithm BESTDIAMSWAP(x, z).

```
BESTDIAMSWAP(x, z).
    if z = x then
    m = 0
              else
                        { z != x }
            if x € D
                        then wait for the information λ(p(x)) from parent.
            end
            L(Tx, z) = max {m, max q € C(z){l(z, q) + height(Tq)}}
             for each local swap edge f = (z, z')  do
            |Pf | = L(Tx, z) + l(f)+ LONGEST(e, f)
|D(Te/f )| = max{|Pf |, |D(T)|}
 end
 for each child q € C(z) do
```

```
    L(Tx \Tq U { f(q, z)}, q) = l(z, q) + max{m, maxs € C(z),s!=q{l(z, s) +        height(Ts)}}
 if x € D then append λ(p(x)) to the enabling information and
            send it to q.
end
if there is no swap candidate then
            candidate whose diameter length is ∞
if z = x then
            store that swap edge as the best swap edge for e
            inform p(x) about the best swap edge.
else
            { z != x }
             send that swap edge to p(z)
end
Algorithm: LONGEST(e = (x, p(x)), f = (z, z0))
if e is on the diameter (i.e., x 2 D(T)) then
            if x € VL then {e € VL }
             if z' € VL then
            return d(z', dc) + |DR|
             else if z' € VC then
            return d(z', dc) + max{|DR|, λ(p(x))}
             else if z' € VR then
            d(dk, ρR) = |DR| - d(ρR, dc)
            d(u', ρR) = |d(u', dc) - d(ρR, dc)|
             d(u', dk) = |DR| - d(u', dc)
d(z', u') = d(z', dc) - d(u', dc)
            from-u' = μR - d(dk, ρR) + d(u', ρR)
            return d(z', u') + max{d(u', dk), d(u', dc) +λ(p(x)), from-u'}
             end
            else {x € VR : symmetric to x € VL }
            else { e not on the diameter }
             if z' € VL then
            return d(z', dc) + |DR|
            else { z' € VC or VL, and |DL| ¸ |DR| }
 return d(z', dc) + |DL|
 end
```

V. CONCLUSION

The point of my paper is to design a distributed algorithm to find out the best swap edges of a minimum diameter spanning tree. A spanning tree T of an undirected graph G is a sub graph that is a tree which includes all of the vertices of G, with minimum possible number of edges. In general, a graph may have several spanning trees, but a graph that is not connected will not contain a spanning tree. If all of the edges of G are also edges of a spanning tree T of G, then G is a tree and is identical to T (that is, a tree has a unique spanning tree and it is itself). A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. Thus we intend on cost reduction which will be economically efficient, remove congestion to avoid any delay of the messages between two communication channels. With the help of above designed algorithm we intend to create a better communication system. We are interested in the designing such an algorithm that will help us in finding out all possible best swap edges by using which we can help create a beneficial system for better communication. And a distributed algorithm is an algorithm, run on a distributed system, which does not assume the previous existence of a central coordinator and a distributed system is a collection of processors that do not share memory or a clock. And therefore, with the use of such an algorithm we will design a system that will help in improving the communication system and make it better.

VI. REFERENCE

[1] [1] B. Gfeller, N. Santoro, and P. Widmayer, "A distributed algorithm for finding all best swap edges of a minimum diameter spanning tree," in Proceedings of 21st International Symposium on Distributed Computing (DISC), ser. Lecture Notes in Computer Science, vol. 4731, 2007, pp. 268–282.

[2] [2] M. Bui, F. Butelle, and C. Lavault, "A Distributed Algorithm for Constructing a Minimum Diameter Spanning Tree," Journal of Parallel and Distributed Computing, vol. 64, pp. 571–577, 2004.

[3] [3] H. Ito, K. Iwama, Y. Okabe, and T. Yoshihiro, "Single Backup Table Schemes for Shortest-path Routing." Theoretical Computer Science, vol. 333, no. 3, pp. 347–353, 2005.

[4] [4] E. Nardelli, G. Proietti, and P. Widmayer, "Finding All the Best Swaps of a Minimum Diameter Spanning Tree Under Transient Edge Failures," Journal of Graph Algorithms and Applications, vol. 5, no. 5, pp. 39–57, 2001,

[5] [5] "Swapping a Failing Edge of a Single Source Shortest Paths Tree Is Good and Fast." Algorithmica, vol. 35, no. 1, pp. 56–74, 2003.

[6] [6] A. D. Salvo and G. Proietti, "Swapping a Failing Edge of a Shortest Paths Tree by Minimizing the Average Stretch Factor," Theoretical Computer Science, vol. 383, no. 1, pp. 23–33, 2007.

[7] [7] P. Flocchini, A. M. Enriques, L. Pagli, G. Prencipe, and N. Santoro, "Point-of-failure Shortest-path Rerouting: Computing the Optimal Swap Edges Distributively," IEICE Transactions on Information and Systems, vol. E89-D, no. 2, pp. 700–708, 2006.

[8] [8] P. Flocchini, L. Pagli, G. Prencipe, N. Santoro, and P. Widmayer, "Computing All the Best Swap Edges Distributively," J. Parallel Distrib. Compute., vol. 68, no. 7, pp. 976–983, 2008.

[9] [9] F. T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays ・ Trees ・ Hyper cubes. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1992.

[10] [10] N. Santoro, Design and Analysis of Distributed Algorithms, ser. Wiley Series on Parallel and Distributed Computing, A. Y. Zomaya, Ed. Wiley, 2007.

[11] [11] N. Santoro and R. Khatib, "Labelling and Implicit Routing in Networks," The Computer Journal, vol. 28, no. 1, pp. 5–8, 1985.

[12] [12] M. Thorup and U. Zwick, "Compact routing schemes," in SPAA '01: Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures. New York, NY, USA: ACM, 2001, pp. 1–10.