

A Logical Topology Based Network Compositor for Packet Distribution System with Effective Edge Replacement

P.R. Nirmal Binu¹, K. Sudharson², N.S. Sivarajan³

¹II ME CSE, S.A. Engineering College Ponthamalle, Chennai -600077, Anna University Chennai,

²Assistant Professor, Department of IT, S.A. Engineering College Ponthamalle, Chennai -600077, University Chennai,

³ME CSE, Sri Muthukumaran Institute of Technology, Chennai – 600069. University Chennai

Abstract---A typical failure in communication networks is disconnection of link between two nodes. When communication link fails, failed links can be replaced by a single swap link. But selecting the best possible link can be very crucial. The edge is chosen from the minimum diameter spanning tree that has been communication overhead and minimizes the time required to adopt routing in case of failure. Spanning tree is based on the labeling and collections of systems in a network using logical topology networks this allows deciding in constant time whether node is leaf node or rooting node using round trip time. The selection of high quality swap links is essential for a reliable communication and it must follow the same objective as the originally chosen communication sub networks. A distributed algorithm that efficiently computes all of these swap links, and how to route messages across swap edges with a compact routing scheme. Spanning tree is based on the labeling and collections of systems in a network using logical topology networks. This allows deciding in constant time whether node is leaf node or rooting node using round trip time. This will minimize the maximum travel time of messages. Hence, each swap link must minimize (among all possible swaps) the diameter of the tree when swapping the edge. We propose a distributed algorithm that efficiently computes all of these swap links, and how to route messages across swap edges with a compact routing scheme. Virtual LAN (VLAN) refers to a group of logically networked devices on one or more LANs that are configured so that they can communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, it is very flexible for user/host management, bandwidth allocation and resource optimization.

Keywords---RTT, Swap Edge MDST, VIAN, SPT, MST.

I. INTRODUCTION

IN an Industrial application that relies heavily on the health of the Ethernet network that attaches all the controllers and computer together, a concern exist about what happens if the network fails. If the result is loss of production or loss of a processed batch or the endangerment of people or equipment, redundancy schemes are examined. Since cable failure is the most likely mishap, cable redundancy is suggested by configuring the network. If one of the segments is lost, then the communication will continue down a parallel path or around the unbroken portion in a topology. The problem with these approaches is that Ethernet supports neither of these topologies without special equipment. This issue is addressed in an IEEE standard numbered 80.1D that covers this standard the concept of spanning tree protocol. A Minimum Diameter Spanning tree

minimizes a largest distance between any pair of node thus minimizing the worst-case length of any transmission path if edge lengths are not uniform. An compact routing scheme for trees which are quickly and inexpensively adapt routing when a failing edge is replaced by a best swap edge. The selection of high quality swap links is essential for a reliable communication and it must follow the same objective as the originally chosen communication sub network. Spanning Tree works by first using an algorithm to find redundant links in the LAN and selecting the best paths. Its initial goal is to put all links in either Forwarding or Blocking. In the end, the links without a redundant link and the best links with a redundant link would be in forwarding state. The redundant links that weren't as good as the selected links would be in blocking state. Spanning Tree cannot use multiple links to the same destination. There is no load-sharing feature with Spanning Tree. Any redundant link that is not as preferred is blocked (essentially shut down) until the primary link goes down.

II. RELATED WORK

In The Distributed algorithm for computing all best swaps of an Minimum Diameter Spanning Tree using no more than $O(n^*+m)$ messages of size $O(1)$ [6]. For shortest path tree an earlier centralized algorithm [6] has been complemented by an distributed algorithm using techniques for finding all best swap edges [1][7]. A Distributed algorithm for computing an Minimum Diameter Spanning Tree in a Graph $G(V,E)$ in an asynchronous setting has $O(n)$ time complexity and uses $O(m,n)$ messages[4]. The algorithm is asynchronous and it works for arbitrary named network topologies. It achieves an efficient $O(n)$ time complexity, and $O(nm \log n + nm \log W)$ bits communication complexity, Where W is the largest weight in the network links[2]. The swap algorithms make use of a set of techniques, which can be tailored on the problem that needs to be solved[5]. In Point-to-point communication model is an undirected graph $G(V,E)$ where V indicates vertex and E indicates Edges corresponding communication links connected between them. Vertices have a unique identifiers and vertices are not aware of topology and edge weight in the entire networks. Communications are carried out in a synchronous manner, Messages are sent at the beginning of each round and received at the end of each round. The model also allows vertices to detect the absence of a message on a link at a given round. The length of a path p in the network is the number of edges it contains. The distance between two vertices u and v is defined as the length of the shortest path connecting them in G . The diameter of G , denoted D , is the maximum distance between any two vertices of G . 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 .

III. System Architecture

In network communication system frequently messages are routed along Minimum diameter spanning tree 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 a new spanning tree. Such an optimal replacing is called a Best Swap, here adjusting an entire tree affects loss in diameter.

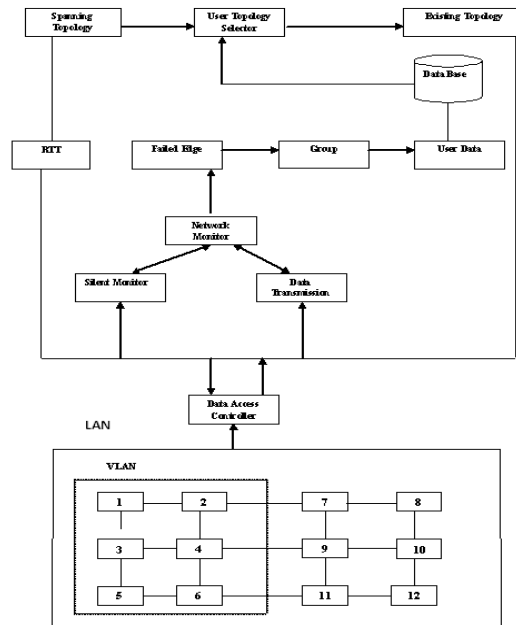


Fig-1: System Architecture

Minimum Diameter Spanning Tree

Minimum Spanning Tree Minimum spanning tree is generated that every node is connected to every node directly or through other edges. Situations where it is desired to explore every node of a graph reached from a given starting node. Problem arises when underlying graph is cyclic, initiating a Depth first search and Breadth first search end up visiting a vertex more than one. Array used to store the information regarding whether a node has already been visited or not. Adding an edge that connects two vertices in a tree creates a unique cycle. Removing an edge from a tree breaks it into two separate sub trees.

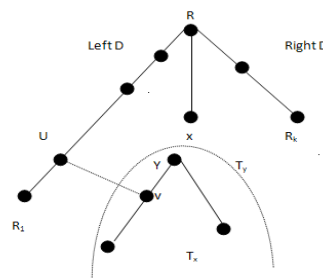


Fig-2: Edge replacement

When edge $e=(x,y)$ is removed then T is divided into two subtrees, T_x and T_y . Which will be later connected by means of a replacement edge $y(u,v)$. $|R(T_{e/g})| = \max\{|R(T_x)|, |R(T_y)|, |P_g|\}$. Different analysis are done depends on whether removed edge is located in diameter area.

A cut of a graph is a partition of its vertices into two disjoint sets. A crossing edge is an edge that connects a vertex in one set with a vertex in the other. Given any cut in an edge-weighted graph (with all edge weights distinct), the crossing edge of minimum weight is in the MST of the graph.

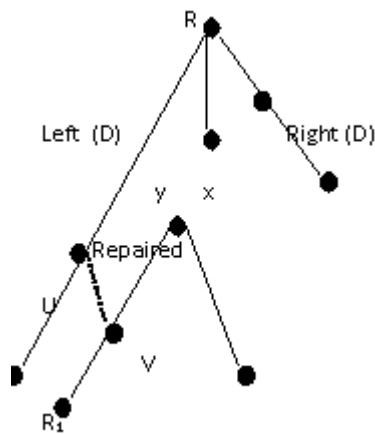


Fig-3: Failed Edge Repaired

IV. EDGE-WEIGHTED GRAPH

Initializing the constructor edge() and weight of the edge and edge Vertex and compare this edge compare() to the string representation Create an empty vertex graph and read graph from input stream number of vertex. Declare v() for number of vertex, e() for edges. **Add edge e to the graph** The **either()** and **other()** methods are useful for accessing the edge's vertices; the **compare To()** method compares edges by weight. If an edge fails, no action is taken as long as no message needs to cross the failed edge. As soon as a message M which should be routed over the failing edge arrives at the point of failure, information about the failure and its best swap is attached to message M , and M is routed towards the swap edge. On its way, all nodes which receive M route it further towards the swap edge, and remember for themselves the information

about the swap. After one message M has been rerouted from the point of failure to the swap edge, all messages originating in the same side of T as M (with respect to the failing edge) will be routed to their destination on the direct path in the tree (i.e., without any detour via the point of failure). If a failing edge has been replaced by a swap edge, then all nodes which know about that swap must be informed when the failure has been repaired. Therefore, a message is sent from the point of failure to the swap edge (on both sides if necessary), to inform these nodes, and to deactivate the swap edge. The above routing scheme has the disadvantage that each node must know the labels of all its neighbors. Thus, an individual node is potentially required to store much more than $O(\log n)$ bits. This drawback could be removed by combining the above scheme with a compact routing scheme for the designer-port model. Such a routing scheme assigns a label of $O(\log n)$ bits to every node, such that the correct forwarding port for a given destination can be computed solely on the basis of the labels of the current position and of the destination. The labels we introduced in our scheme are then only used to determine whether a message needs to be rerouted (because it would otherwise need to use the failing edge). As this is possible solely on the basis of the labels of the message's current position and its destination, this combination yields a compact routing scheme which can efficiently adapt to swaps.

A. Forming Spanning Tree

Spanning trees job is to find all links in the network and shutdown any redundant ones, thereby preventing network loops from occurring. Spanning tree does this by electing a root bridge that will preside over a network topology decisions. Spanning tree creation unit creates a plurality of spanning tree based on the network topology, and register the spanning trees to a spanning tree information table. Spanning tree selection unit selects a spanning tree where in communication resources will be used most effectively, spanning tree information table.

A node setup unit sets the mapping information which express correspondence between requested Virtual LAN and the selected spanning tree in each node. Link priority setting unit for adding link priority level information which express the priority level of each of the link between the pluralities of node to the topology information stored is said to be topology information storage.

The VLAN creation device set link priority setting unit adds the link priority level information of each of the links based on instruction from network administrator. Link priority setting unit adds the link priority level information of the links based on bandwidth of the each link. A spanning tree is a graph where all the nodes in the graph are connected in some way, with the

requirements that there are no cycles in the graph. The idea is to build the spanning tree in parallel and merge more and more of the smaller trees in to bigger ones and finally in the construction phase that will be formed into a single spanning tree. When node receive the message to start computing the spanning tree, they define themselves as the root to their one-leaf big spanning tree and set their adjacent list according to the edges they have to their neighbor.

B. Network Monitor

In spatial topology network the work of network monitor is to monitor the network configuration in a LAN. The Network Monitor manages the in and Out Data Bound from and to the system buffer or stack at primary transport level. This module uses the UDP (User Datagram Protocol) for identifying the systems that are active in present time. It works as the Main module which manages the LAN Access and Data Transmission for the specific systems in Network Tree. This module also takes look of the VLAN Created by the Spanning Tree Network.

C. Silent Monitoring and Data Transmission

Silent monitoring works as a background process for monitoring the systems that changes its state i.e. off state to on state or vice versa. It looks out of the systems currently in the network and goes in out of network as temporarily or fully. And also notes the systems that join the network after forming the spanning tree. This module keeps on updating the systems currently in the VLAN. This module can run as background process or active process. This module runs under Daemon process which closes all the background work as the application shuts down.

D. Topology Grouping and Tree Forming

It collects the data or system details from the Silent monitoring system and makes the system or network group based on the Round Trip Time of the network system. This system grouping can be changed at any level based on the silent module. This module forms the spanning tree network for creating a logical topology for the VLAN configured by the spanning tree topology.

E. LAN Access Controller

Access controller is the core module which blocks or allows data from the VLAN. It uses the C++ Lower level language to filter data the transport layer of the operating system. This module also checks the shortest route from the source to destination system. This Need to be run in system tray for making the VLAN more secure and robust. The VLAN access controller blocks data from the outer LAN to the VLAN. This

module also blocks the any kind of protocol used by the outer LAN.

F. Topology Viewer

The topology viewer arranges the system in a logical network based on the topology selected. Which can be Star, Bus, Hybrid or spanning tree based network. This Module arranges the systems based on the topology. The primary use for this field is to allow management functions (such as the LAN Network Manager product for monitoring and managing networks)to monitor the performance of the spanning-tree protocol.

V. THE BEST SWAP ALGORITHM

For a given edge $e = (x, p(x))$ that may fail, each node z in the sub tree T_x rooted at x executes the following steps:

- Wait for the enabling information from the parent (unless $x = z$), and then Compute the enabling information for all children and send it For each local swap edge $f = (z, z')$, compute $|D(T_{e/f})| = \max\{(D(T)_i, |L(T_{x,z})| + l(f) + |L(T_{x,z'})|)\}$
- From each child $q \in C(z)$, receive the node label of a best swap edge candidate and its resulting diameter.
- Choose the best swap and send to parent.
- No swap edge creates a dummy candidate whose diameter.

V. EXPERIMENTAL RESULT

The figure 4. shows the Performance of LAN and VLAN in terms of time and number of systems connected in a network.

- Introduce Network monitoring it monitors the edges.
- Replaces the failed edge immediately and use the current path.
- Compact Routing scheme used in routing table.
- Updates the topological changes.
- User can select the topology from the list, based on the

Selected topology logical network will be formed. Spanning topology forms the spanning tree; it prevents loops in the VLAN and to select the fastest network links.

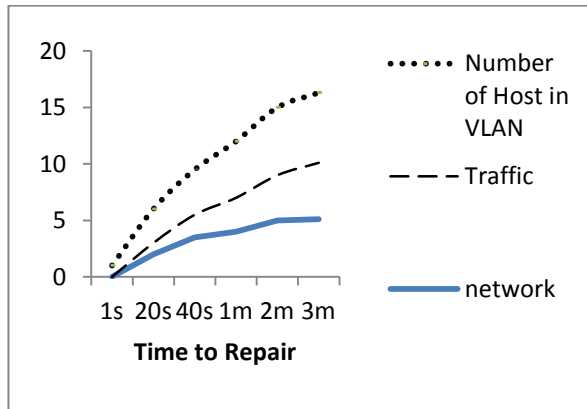


Figure 4: Performance of VLAN

VI. CONCLUSION AND FUTURE ENHANCEMENT

In this paper the logical topology based network compositor for packet distribution system with effective edge replacement we proposed that a typical failure in communication networks is disconnection of link between two nodes. When communication link fails, failed links can be replaced by a single swap link. Implementation in virtual LAN dynamically based on information such as the source MAC address of the device connected to the port or the username used to log onto that device. User has an optional to select the existing topology. Topology viewer, monitoring and forming a Virtual group reducing the time and space complexity in virtual local area network. In future work data can be transmitted from VLAN to WAN to reduce the time complexity. Also to transmit secured data from WAN to LAN.

ACKNOWLEDGMENT

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. First, I thank my Internal Guide **Mr. K.Sudharson M.E**, Assistant Professor Department of IT as well as Head of the Department **Mrs.UMARANI SRIKANTH M.E. (Ph.D.)** Department of Computer Science and Engineering, without her guidance and patience, this dissertation would not be possible. I also wish to record my thanks to project panel members, Professors of the Department of Computer Science and Engineering for their consistent encouragement and ideas.

REFERENCES

- [1] 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.

- [2] P. Flocchini, A. Mesa Enriquez, L. Pagli, G. Prencipe, and N. Santoro, "Point-of-Failure Shortest-Path."
- [3] Bonaventure, O., Filsflis, C., and Francois. P Achieving Sub-50 Milliseconds Recovery upon BGP Peering Link Failures. *Communication Complexity (SIROCCO'94)*, Carleton Unprocess, pages 99-109, 1995.
- [4] D.Passmore and J. Freeman, " The Virtual LAN Technology Report 3COM White paper , 1996.
- [5] E. Nardelli, G. Proietti, and P. Widmayer, "Finding All the Best Swaps of a Minimum Diameter Spanning Tree under Transient Edge Failures," *J. Graph Algorithms and Applications*, vol. 5, no. 5, pp. 39-57, 2001.
- [6] E. Nardelli, G. Proietti, and P. Widmayer, "Swapping a Failing Edge of a Single Source Shortest Paths Tree Is Rerouting: Computing the Optimal Swap Edges Distributively," *IEICE Transactions*.
- [7] Gfeller, N. Santoro, and P. Widmayer, "A Distributed Algorithm for Finding All Best Swap Edges of a Minimum Diameter Spanning Tree," *Proc. 21st Int'l Symp. Distributed Computing (DISC)*, pp. 268-282, 2007.
- [8] 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.
- [9] M. Bui, F. Butelle, and C. Lavault, "A Distributed Algorithm for Constructing a Minimum Diameter Spanning Tree," *J. Parallel and Distributed Computing*, vol. 64, pp. 571-577, 2004.
- [10] M. Thorup and U.Zwick, Compact routing schemes, In *SPAA 2001*, 1-10.
- [11] P. Flocchini, L. Pagli, G. Prencipe, N. Santoro, and P. Widmayer, "Computing All the Best Swap Edges Distributively," *J. Parallel and Distributed Computing*, vol. 68, no. 7, pp. 976-983, 2008.
- [12] R.Tan and J. van Leeuwen. Compact routing methods: a survey . In *proc. 1st Coll. On Structural information and Jens Krause.*, Automatic VLAN creation based on-line measurement, 2006.
- [13] Y. Afek, M. ricklin. Sparser: a paradigm for running distributed algorithms. *Journal of Algorithms*, 14:316-328, 1993.

Biography



Sudharson Kumar received a Master Degree in Computer Science and Engineering from S.A.Engineering College, Anna University, Chennai. He is working as Asst. Prof. in Department of Information Technology, S.A. Engineering College Chennai. He has awarded Best Outgoing Student Award while doing his Master Degree, and he started his research in Adhoc Network while pursuing his Masters Degree and he presented more than 10 papers in various National and International Conferences. He is member of various societies like IACSIT, IAENG etc., He published various international journals.



Nirmal Binu P.R Pursuing Master of engineering in Computer science at S.A. Engineering College Anna University of technology Chennai. He has received his Master in Computer Applications degree from Noorul Islam College of engineering, Anna University, Chennai.



International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 1, March 2012



N.S.Sivarajan Pursuing Master of engineering in Computer science at SriMuthukumaran College of Engineering Anna University of technology Chennai. He has received his Master in Computer Applications degree from Noorul Islam College of engineering, Anna University, Chennai.