2011 Workshops of International Conference on Advanced Information Networking and Applications

Network Failure Recovery with Tie-Sets

Kiyoto Kadena, Kiyoshi Nakayama, *Student Member, IEEE*, and Norihiko Shinomiya, *Member, IEEE*
Graduate school of Engineering, Soka University, Tokyo, 192-8477, Japan
shinomi@ieee.org

***Abstract*— The present research aims to recover network failure in networks with a complicated topology by focusing on tie-sets. A tie-set implies a set of links constituting a loop. The entire network is divided into smaller local units of tie-sets which encompass all network vertices and links. These units realize network management for quick and flexible failure recovery. This paper first introduces the concept of tie-sets, and then proposes a recovery method for single link failures. The latter half of the paper deals with the efficiency of failure recovery in different spanning trees. Since a hypothesis is formed by a numerical evaluation the paper proposes a method of determining a spanning tree for quick link failure recovery.**

1. Introduction

Owing to quick and remarkable developments in information technology, the Internet has spread dramatically across the world and become a useful tool in daily life. Due to an increasingly large and complicated structure of networks, a local and flexible controlling method to restore network failures is becoming increasingly necessary.

Ring-based restoration leads to reliable and quick link failure recovery, since this method makes it possible to restore a failure with only one path shifting from a failed path to a backup path. Ethernet Automatic Protection Switching (EAPS) is a protocol focusing on a ring configuration and is used in local area networks [1]. However, this protocol can be applied to only ring or multi-ring topology networks, not to complicated networks. Link Aggregation Control Protocol (LACP) is a protocol which assures reliable communication by multiplexed paths [2]. Paths multiplexed are used as backup paths and a failed path is quickly shifted to a backup path. To sustain a network with this protocol, a number of ports are needed. Therefore, the redundant efficiency is not as high as that of other protocols.

In recent years, mesh topology networks have been more focused. This configuration enables high flexibility in routing management. Taking advantage of this merit, contemporary networks tend to have mesh-type topology. This trend is expected not to change for years to come. Although efficient failure recovery is possible in mesh networks, a defect comes from the multiplicity of links available for restoration.

An improved version of Spanning Tree Protocol (STP), called Rapid Spanning Tree Protocol (RSTP), is the most commonly used protocol in mesh topology networks today [3]. RSTP is run with a spanning tree and is used to prevent a network from broadcast storms. RSTP constructs the shortest paths based on the communication speed or link cost and realizes fast route shifts to recover from link failures. A weakness of RSTP is that there is a possibility one link failure will affect the entire network and cause a longer convergence time in a recovery. Currently, RSTP is applied to ring configurations in existing networks to fulfill a quick recovery.

As stated earlier, the protocols utilizing rings in mesh networks already exist. Since local management of networks focusing on rings are based on mathematical theory and unit distribution control their use is highly complex. In order to overcome this complexity, the present research overcomes this complexity by proposing a simplified method using tie-sets. In graph theory, a tie-set is defined as a set of links constituting a loop. An entire network can be divided into a set of tie-sets. Tie-sets make it possible to manage an entire network as an autonomous distribution system by focusing on each tie-set [4] [5]. This paper also discusses the efficiency of link failure recovery using tie-sets. It further introduces a way to determine a fundamental system of tie-sets for quick link failure recovery.

978-0-7695-4338-3/11 $26.00 © 2011 IEEE
DOI 10.1109/WAINA.2011.125

2. Tie-set and state information

A. Fundamental system of tie-sets

A tie-set is defined as a set of links creating a logical loop, and a fundamental system of tie-sets represents a set of all tie-sets in a network. A fundamental system of tie-sets covers all vertices and links in a network [6] [7]. In a given connected and undirected graph $G = (V, E)$ with a set of vertices $V = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $E = \{e_1, e_2, \ldots, e_m\}$, let $L_i = \{e_1^i, e_2^i, \ldots, e_k^i\}$ be a tie-set in G and $L = \{L_1, L_2, \ldots, L_k\}$. A set of tie-sets in a network is called a "fundamental system of tie-sets." Additionally, let T and e_k ($\in T$) be a tree of G and a failed link respectively. In this research, L^f ($\subset L$) represents a set of tie-sets which meets $e_k \in L_i$ ($L_i \in L$) Therefore, the number of tie-sets in which e_k is contained is defined as $|L^f|$. Fig.1 (a) gives an example of fundamental system of tie-sets denoted by L1 through L4 in a graph. Each tie-set contains links as shown in formulas.

As a principle, once a spanning tree is determined, only one fundamental system of tie-sets will be automatically chosen. A spanning tree represents a sub-graph which covers all links and vertices in a single network with no loops. In order to create a spanning tree, Breadth First Search (BFS) and Depth First Search (DFS) are generally used. Based on conclusions drawn from earlier researches conducted, BFS is more suitable searching methods for quick failure recovery. This therefore is the rationale in the present research to create a spanning tree based on the BFS method.

B. State information of each vertex

Each vertex *N* takes hold of three types of state information as below.

(1) Incident links: Information of links which are connected to *N*

(2) Adjacent vertices: Information of vertices which are connected to *N*

(3) Tie-set Information: Information of tie-sets to which *N belongs*. In case a tie-set $L_i \ni N$, *N* belongs to L_i and takes hold of information of L_i.

For instance, in Fig.1 (a), vertex v2 takes hold of information of {e2, e3, e5, e6} as incident links, {v0, v1, v3, v4} as adjacent vertices, and $\{L_1, L_2, L_3\}$ as tie-set information.

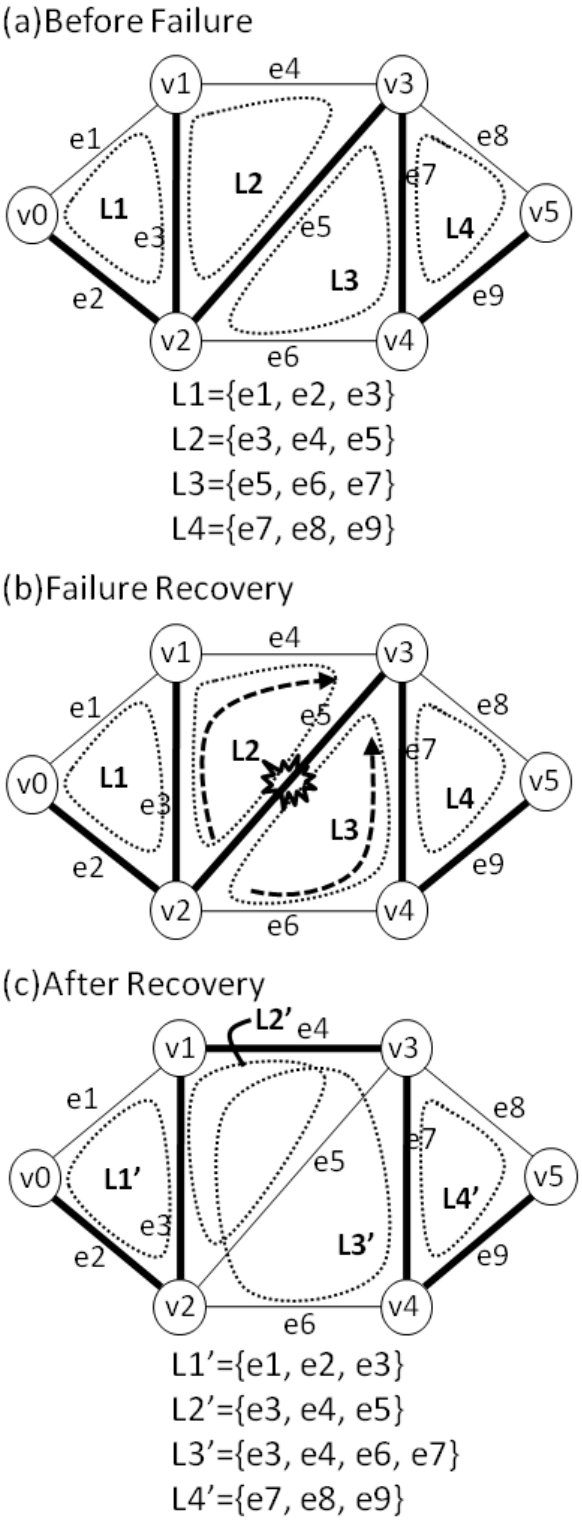

Fig. 1 : Behavior in link failure recovery

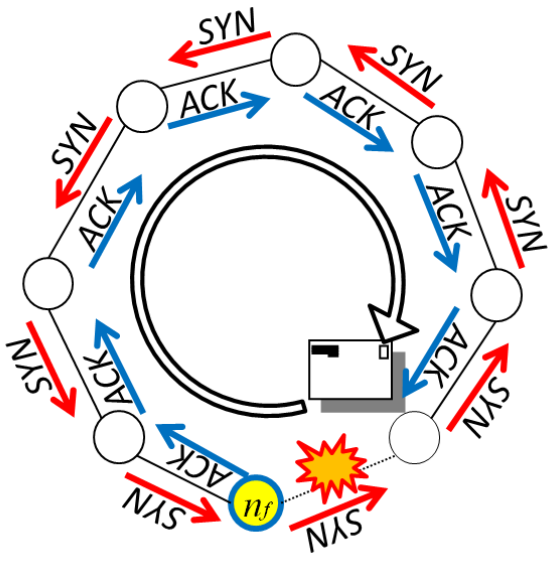

Fig. 2 : A tie-set agent in a loop

C. Failure detection in a tie-set

Tie-set agents or messages that are regularly circulating on each tie-set, are utilized to detect failure in information network links. Figure2 shows a way to detect a single link failure in a tie-set. The tie-set agent circulating on tie-set L_i holds the state information of all vertices that belong to L_i. If the state information changes anywhere in L_i, the tie-set agent is able to understand the change and subsequently informs other vertices in the tie-set of such a change taking place. Once a link failure occurs, the agent stops circulating.

Each time the agent goes through a vertex in L_i, the vertex sends a SYN message to an adjacent vertex in a direction opposite to that of the tie-set agent in L_i. If the adjacent vertex receives an ACK message that notifies it that the agent is moving normally, it understands that the system is working properly. On the other hand, if the adjacent vertex does not receive an ACK message, it understands that a failure has taken place in the link between these two vertices. In such an event the last vertex will negotiate with other vertices and identify a vertex which will then conduct the failure recovery.

3. Link failure recovery based on a tie-set

Figure 1 (a), (b) and (c), where communication links are represented as thick/thin lines and backup paths, best describe the behavior of information networks during a link failure recovery based on a tie-set. As mentioned earlier, each vertex seizes the information of tie-sets to which the vertex belongs. In order to share tie-set information within the minimum number of loops, each vertex conducts a distributed algorithm.

As shown in Figure 1 (c), after failure recovery, the original system of tie-sets changes to a new system of tie-sets, denoted by L1' through L4', that use e4 as an alternate path. Whenever a failure occurs in one link, the damaged link is quickly replaced by an alternate link to maintain communication in the network.

The present tie-set method updates network state information on each vertex through transmitted messages from the detecting vertex to other vertices. The transmitted messages include the state information of the new network and are then transmitted along the original loops during the process of replacing the damaged path. For instance, in Figure 1 (b), when vertex n2 detects a link failure on e3, the new network state sends messages regarding its new configured tie-sets to L2 and L3—the two tie-sets associated with link failure. Each vertex receives the messages and understands to which tie-sets it belongs.

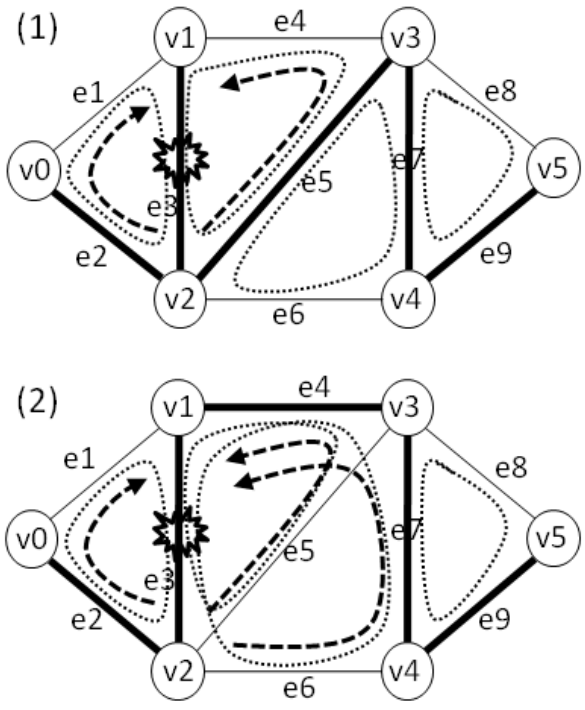

Fig. 3 : Restoration procedures in different spanning trees

4. The fundamental system of tie-sets and failure recovery efficiency

In general, it is possible to have more than one spanning tree in a given network structure. The network structure is organized in such a manner that it supports failure recovery. A failure recovery is processed through loops containing a damaged link which is a part of the network structure. Therefore, a fundamental system of tie-sets that is determined by a spanning tree further enhances the efficiency of link failure recovery in a network. This hypothesis and process are validated in Figure 3 which shows two systems of tie-sets constructed from the same graph.

Figure 3 (1) and (2) show the same numbers of tie-sets, but the process of failure recovery in both systems are quite different. In Figure 3 (1), the failed link belongs to two tie-sets. On the other hand, in Figure 3 (2), it belongs to three tie-sets. For example if vertex n2 detects a failure on link e3, it transmits messages along each loop as shown in Figure 3 (1) and (2). More messages are needed to restore a link failure than as demonstrated in Figure 2 (1). Thus, the spanning tree in Figure 3 (1) is expected to restore link failure more effectively and quickly than the spanning tree in (2).

5. Numerical evaluation

In order to estimate the efficiency of link failure recovery in different fundamental systems of tie-sets, a numerical evaluation is conducted. Fourteen spanning trees, T1 to T14 are output on the same network structure, and the relationship between each spanning tree and failure recovery efficiency is investigated. As evaluation indicators, (1) the number of messages and (2) the number of affected vertices during failure recovery are used. We assume a link failure on each link of all of the spanning trees and measure the average values of the two indicators. In this evaluation, lower values of these indicators are expected to result in quicker failure recovery.

Table 1 and Fig. 4, corresponding to the table, show the result of this evaluation. As seen from the results, T1, the spanning tree that will lead to the most efficient failure recovery, enables failure recovery with less than half the numbers of both messages and affected vertices for T3, a spanning tree that will lead to inefficient failure recovery.

Fig. 5 shows the distributions of tie-sets in these two spanning trees. In T3, links are contained in relatively more tie-sets than in T1. In a restoration based on a tie-set, more procedures are required for failure recovery in T3 than those required in T1. This research focuses on the maximum value of the number of tie-sets in which a link is contained and tries to find a spanning tree to minimize this value.

Table 1 : Evaluation of spanning trees

	Variations of Spanning Trees													
Index	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
# of messages	1.78	2.33	3.78	2.22	2.44	3.11	2.22	2.00	2.33	2.00	2.00	2.89	2.78	2.67
# of affected vertices	4.11	5.67	8.33	5.67	6.56	7.00	5.89	5.00	5.44	5.00	5.00	7.00	9.00	7.44

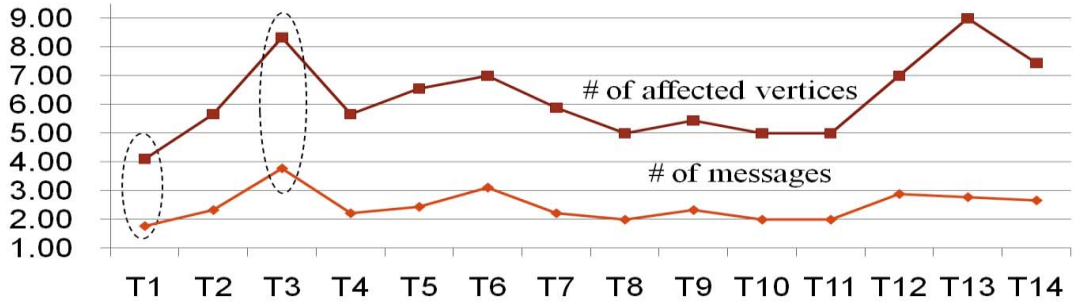

Fig. 4 : Evaluation of spanning trees

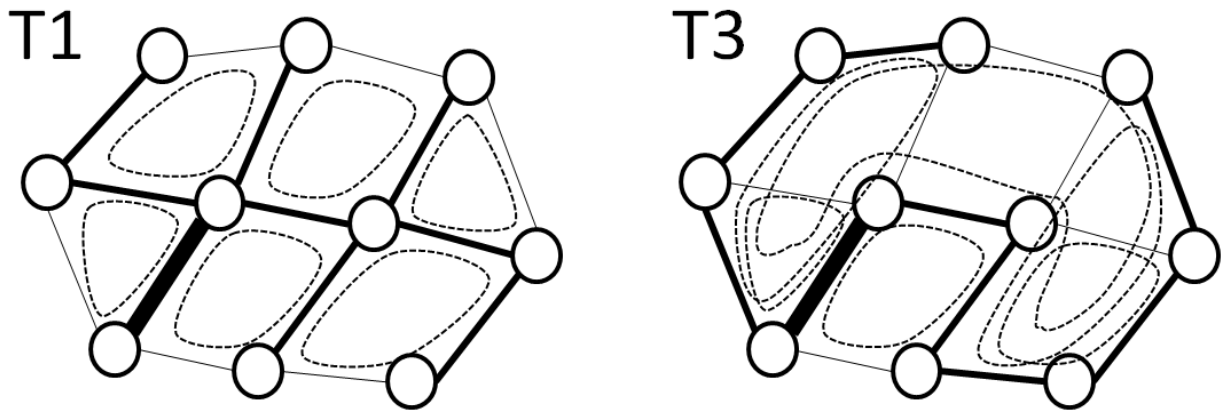

Fig. 5 : Distributions of tie-sets in T1 and T3

6.Discussion on a determining method of a spanning tree

As mentioned before, failure recovery efficiency depends on the fundamental system of tie-sets, and a fundamental system of tie-sets is gained by a spanning tree. Thus, quicker failure recovery becomes possible by choosing a more appropriate spanning tree. This section discusses a determining method of a spanning tree as below.

Step1:

In order to output a spanning tree as an initial solution, firstly, the vertex, which is connected to the greatest number of vertices, is chosen as a root vertex. The vertex becomes the origin of the spanning tree. Then, a spanning tree is created based on BFS.

Step2:

In this phase, a new spanning tree is created by shifting from one used edge to one backup edge in the current spanning tree. Then, a new spanning tree is compared with the current tree. These two spanning trees are evaluated using the value shown below:

$$C_{Max} = \max_{e_k \in T} |L^f|$$

If the value C_{Max} for a new tree is smaller, the current tree is disposed of and the new tree is updated. Fig.6 shows an update from the tree (A) to the tree (B). The tree (B) is gained by shifting from one used edge to one backup edge on the tree (B). The tree (B) makes C_{Max} smaller than the tree (A) does. Thus, the new tree (B) becomes a new solution. This step is repeated unless there are no further spanning trees which can be created from the current spanning tree.

Step 3:

The current tree is the optimal spanning tree found based on the value C_{Max}. Finally, this spanning tree is output as the optimal solution.

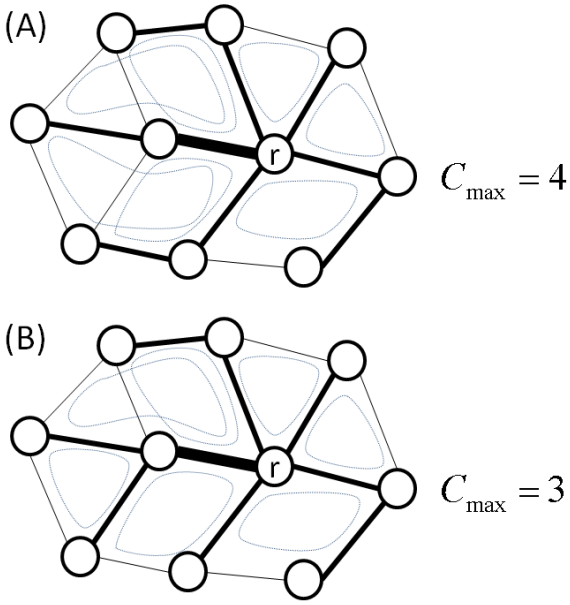

Fig. 6 : Shifting to a new edge

7.Conclusion

In conclusion the paper has attempted to do two things: it first proposes a link failure recovery method and second it determines a spanning tree to restore link failure. It further proposes the future direction of this research. Elaborating on the first point, the paper proposes a link failure recovery method with tie-sets. By focusing on rings, called tie-sets, in a network and controlling them as local units, flexible network management for link failure recovery becomes possible. Explaining the second point this research discusses the failure recovery efficiency and introduces a way of determining a spanning tree for link failure. This method does not guarantee that a spanning tree to optimize the given value is output as a final solution. However, using this method, the network can be managed more effectively with more appropriate spanning trees than those chosen at random. Finally the research proposes the development of a simulator which incorporates the functions based on the proposed method is required. In addition, it is also necessary to examine whether this method can be made to work in practical network management through the use of a simulator.

References

[1] "Application Notes for Extreme Networks Ethernet Automatic Protection Switching (EAPS) with Avaya Communication Manager and Avaya P330 and C360 Stackable Switches - Issue 1.0," http://support.avaya.com/css/P8/documents/003840119

[2] "Configuring LACP (802.3ad) Between a Catalyst 6500/6000 and a Catalyst 4500/4000," http://www.cisco.com/en/US/tech/tk389/tk213/technologies_configuration_example09186a0080094470.shtml

[3] "IEEE 802.1D-2004 IEEE Standard for Local and Metropolitan Area Networks -- Media access control (MAC) Bridges," http://standards.ieee.org/getieee802/802.1.html

[4] N.Shinomiya, et al., "A theory of tie-set graph and its application to information network management," J. Circuit Theory and Applications, Vol.29, No.4, pp.367-379, Jul. 2001.

[5] T.Koide, et al., "A study on the tie-set graph theory and network flow optimization problems," J. Circuit Theory and Applications, Vol.32, No.6, pp.447-470, Nov. 2004.

[6] A.Bondy and U.S.R.Murty, Graph Theory, Springer: 2007.

[7] K.Nakayama, N.Shinomiya and H.Watanabe, "Distributed Control for Link Failure Based on Tie-Sets in Information Networks," IEEE International Symposium on Circuits and Systems (ISCAS), Proc. pp.3913-3916, Paris France, May 2010.