# Autonomous Recovery for Link Failure Based on Tie-Sets in Information Networks

Kiyoshi Nakayama, Norihiko Shinomiya
Graduate School of Engineering, Soka University, Tokyo 192-8577, Japan
Email: {kiyoshi.nakayama, shinomi}@ieee.org

***Abstract*—This study proposes an autonomous distributed control method for single link failure based on loops in a network. This method focuses on the concept of tie-sets defined by graph theory in order to divide a network into a string of logical loops. A tie-set denotes a set of links that constitutes a loop. If tie-sets are used as local management units, high-speed and stable fail-over can be realized by taking advantage of ring-based restoration. This paper first introduces the notion of tie-sets, and then describes the distributed algorithms for link failure. Experimental results comparing the proposed method with RSTP suggest that our method alleviates the influence of failure in terms of route switching points, adverse effect in communications, and recovery time with a modest increase in state information of a node.**

***Index Terms*—graph theory, tie-set, loop, link failure**

## I. Introduction

As the internet continues to grow in size and complexity, it is essential to manage information networks locally and flexibly with autonomous distributed control architectures. In modern networks becoming larger and more complicated, such failure, even for a short time, may cause extensive damage to entire network lines. For this reason, high-speed and reliable restoration of network failures becomes especially important.

It is known that ring-based restoration can realize high-speed and stable fail-over because of the availability of exactly one backup path between any two nodes, leading to simple automatic protection switching mechanisms. For instance, Unidirectional Path Switched Ring (UPSR) [1] or Bidirectional Line Switched Ring (BLSR) [2] is used in a Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) network. Moreover, Ethernet Automatic Protection Switching or EAPS [3] is utilized in local area networks. In ring restoration, in case of link failure, the end-nodes of the link switch to the backup path joining the two end-nodes. In path protection, all affected connections are notified of the link failure, and they switch to the backup paths. However, the route switching technique in rings requires the reservation of half of the total capacity for protection purposes.

More recently, attention has focused on mesh networks partly because of the increased flexibility they provide in routing connections, and partly because the natural evolution of network topologies leads to a mesh-type topology. While failure recovery in mesh networks can potentially be more efficient, it is more complex as well because of the multiplicity of routes which can be used for recovery.

Today, Rapid Spanning Tree Protocol (RSTP) [4] is generally used for mesh networks to solve the problem of traffic loops and broadcast storms. RSTP is an evolution of the Spanning Tree Protocol (STP), and introduced to provide faster spanning tree convergence after a topology change to reduce recovery times. A major issue of RSTP is that the additional complexity of meshed topology causes fail-over times to increase in the vicinity of a root bridge [5].

The proposed method focuses on the concept of tie-sets, which is defined by graph theory, in order to divide a mesh network into a set of $\mu$-dimensional logical loops. A tie-set denotes a set of all links that constitutes a loop. Based on theoretical rationale of graph theory, a string of tie-sets that can cover all the nodes and links of a network is created by utilizing a spanning tree, even in a nonplanar mesh network. There have been previous works that focus on tie-sets in a mesh network, so called a "bi-connected graph" in graph theory [6], [7]. Those studies analyze graph theoretical nature of underlying loops in a network as well as indicate possibilities of conducting optimal local network management based on tie-sets. An overview of fault link avoidance based on tie-sets is also suggested in [8], [9]. Particularly based on the notion of a "fundamental tie-set", route switching can instantly be realized by shifting a failed path to a non communication path that uniquely exists in a fundamental tie-set. Thereby, ring protection is virtually realized on a logical loop defined by a tie-set. As a result, adverse effects for communications caused by route switching are greatly reduced compared with STP or RSTP due to the local control in a logical ring formed by a tie-set. In this paper, we try to analyze the strengths of the proposed method from a perspective of distributed algorithms [10] as well as experiments.

## II. Tie-sets and State Information

### A. Fundamental System of Circuits and Tie-sets

For a given bi-connected and undirected graph $G = (V, E)$ with a set of vertices $V = \{v_1, v_2, ..., v_n\}$ and a set of edges $E = \{e_1, e_2, ..., e_m\}$, let $L_i = \{e_1^i, e_2^i, ..., e_k^i\}$ be a set of edges which constitutes a loop in $G$. The set of edges $L_i$ is called a "tie-set" [11]. Let $T$ and $\overline{T}$ be a tree and a cotree of $G$, respectively, where $\overline{T} = E - T$ [12]. $\rho = \rho(G) = |T|$ and $\mu = \mu(G) = |\overline{T}|$ are called the $rank$ and the $nullity$, respectively. A tree $T$ on a graph $G = (V, E)$ is an ultranet set of edges which does not include any tie-set. In other words, for $l \in \overline{T}$, $T \cup \{l\}$ includes one tie-set. Focusing on a subgraph

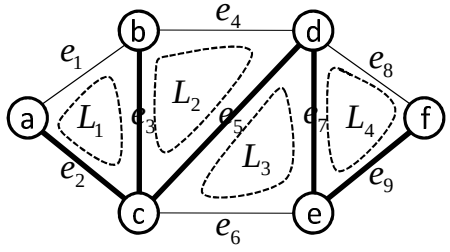


Fig. 1. An example of a fundamental system of tie-sets

$G_T = (V, T)$ of $G$ and an edge $l = (a, b) \in \overline{T}$, there exists only one elementary path $P_T$ whose origin is $b$ and terminal is $a$ in $G_T$. Then an elementary circuit that consists of the path $P_T$ and the edge $l$ is uniquely determined as follows:

$$\begin{aligned} L(l) &= (a, l = (a, b), P_T(b, a)) \\ &= (a, l, v_0 = b, t_1, v_1, \cdots, t_h, v_h = a) \end{aligned} \quad (1)$$

In this way, a circuit determined by an edge $l = (a, b) \in \overline{T}$ and a path $P_T(a, b)$ on $G_T$ is denoted as a "fundamental circuit". A simple circuit can be expressed by a set of edges, so called a tie-set. A tie-set corresponding to a fundamental circuit regarding $T$ is denoted as a "fundamental tie-set" regarding $T$. It is known that $\mu$ fundamental circuits and tie-sets exist in $G$, and they are called a "fundamental system of circuits" and a "fundamental system of tie-sets", respectively. A fundamental system of tie-sets covers all the vertices and edges in $G$ as shown in Fig.1.

### B. State Information of a Node

Each node $n$ mainly has three types of state information as follows:

*1) Incident Links:* Information of links connected to $n$.

*2) Adjacent Nodes:* Information of nodes which are connected through incident links of $n$.

*3) Tie-set Information:* Information of fundamental tie-sets to which $n$ belongs. When a fundamental tie-set $L_i \ni n$, it is defined that $n$ belongs to $L_i$ and has information of $L_i$.

Here is an example of state information of a node $c$ in Fig.1. The node $c$ has information of $\{e_2, e_3, e_5, e_6\}$ as incident links, $\{a, b, d, e\}$ as adjacent nodes, and tie-set information of $\{L_1, L_2, L_3\}$.

### C. Algorithm for Configuring Tie-set Information

In order to obtain Tie-set Information, each node executes a distributed algorithm to recognize fundamental tie-sets. By automatically configuring state information of tie-sets, a burden for initialization of network managers is greatly lessened especially in large-scale networks.

In information networks, let us assume that a tree $T$ on $G = (V, E)$ corresponds to communication paths, and a cotree $\overline{T}$ corresponds to non communication paths. A tree $T$ is easily constructed by executing the spanning tree algorithm (STA), which is one of the basic distributed algorithms. In this paper, links representing communication paths are defined as "tree links" which are expressed by thick lines, while links representing non communication paths are defined as "cotree links" which are expressed by thin lines as shown in Fig.1. For example, tree links and cotree links are $\{e_2, e_3, e_5, e_7, e_9\}$ and $\{e_1, e_4, e_6, e_8\}$ in Fig.1, respectively.

A *Find Tie-set* message, which is used to catch information of a fundamental tie-set, includes information as follows:

- *EdgeTable:* A set of links through which a *Find Tie-set* message passed
- *NodeTable:* A set of nodes through which a *Find Tie-set* message passed

If *Find Tie-set* messages are processed according to the rules below, each node can hold information of fundamental tie-sets. First, each node $n_o$ creates *Find Tie-set* messages, and then sends those messages to all adjacent nodes of $n_o$. When sending a *Find Tie-set* message to an adjacent node $n_a$, $n_o$ adds node information of $n_o$ to *NodeTable*, and adds information of a link connected to both $n_o$ and $n_a$ to *EdgeTable*. Let $n_r$ be a node which receives a *Find Tie-set* message. After receiving a *Find Tie-set* message, $n_r$ executes different procedure by the following cases.

*Case 1:* $n_r \neq n_o$ In this case, if *EdgeTable* of the *Find Tie-set* message includes more than one cotree link, $n_r$ discards the message. If *EdgeTable* contains no or one cotree link, $n_r$ copies the *Find Tie-set* message and sends the copied message to adjacent nodes which are not included in *NodeTable*. In case that the adjacent node is $n_o$, $n_r$ sends the copied message to $n_o$ even if $n_o \in$ *NodeTable*. When sending a copied message to an adjacent node $n_a$, $n_r$ adds node information of $n_r$ to *NodeTable*, and adds information of a link connected to both $n_r$ and $n_a$ to *EdgeTable*.

*Case 2:* $n_r = n_o$ In this case, the *Find Tie-set* message has passed through certain loop in a network. If *EdgeTable* coincides with a fundamental tie-set, the information of *EdgeTable* and *NodeTable* included in the *Find Tie-set* message is stored in $n_o$.

In this algorithm, the number of messages casted on a network, so called Communication Complexity, can be analyzed by focusing on a format of the *Find Tie-set* message. The communication complexity is determined to be $O(n^4)$ from a perspective of distributed algorithm, where $n = |V|$. As for execution time, a message passes through on tree links and one cotree link. Therefore, time complexity is $O(D)$ where $D$ is defined as a diameter of a graph $G$.

## III. Distributed Control for Link Failure

### A. Procedure of a Node in Failure Detection

*1) Distributed Algorithm for Link Failure:* When a link failure occurs in network lines, a node connected to the failed link detects the failure. If two nodes are connected to the failed link, and the both of them detect the failure at the same time, the node which has a smaller address takes the responsibility to restore the failure. Let $n_f$ be a node which detects a link failure and $e_f$ be a failed link. The procedure in $n_f$ is listed as follows:

Step1 *Blocking physical ports connected to* $e_f$
$n_f$ blocks its physical port connected to $e_f$, and

Fig. 2. Behavior in changing a communication path

sends a *Close Port* message to another node which is connected to $e_f$. Then the node blocks its physical port connected to $e_f$.

Step2 *Choosing a tie-set $L_i$ from Tie-set Information, where $e_f \in L_i$*
$n_f$ chooses tie-sets which include $e_f$ from its Tie-set Information.

Step3 *Determining a tie-set $L_r$ to conduct route switching*
When several tie-sets including $e_f$ exist in $n_f$, $n_f$ chooses one tie-set $L_r$ in which the communication path is shifted.

Step4 *Opening physical ports connected to the cotree link*
$n_f$ sends an *Open Port* message to the nodes which are connected to the cotree link of $L_r$ to resume communication.

In Step 3 above, there are several criteria to decide a tie-set in which route switching is conducted. Criteria are, for instance, the number of hops to cotree link, the total link cost of a tie-set, and the size of a tie-set, etc. It depends on the characteristics of a network which tie-set is appropriate to shift a communication path. If there is not any unique feature in a network, the number of hops becomes a proper criterion. The behavior of the procedure above is shown in Fig.2.

*2) Procedure after Link Failure:* Next procedure after the steps above depends on the kind of link failure. There are mainly two kinds of link failure. One is that link failure can be restored. Another is that link failure cannot be restored permanently or a failed link itself is removed.

Fig. 3. Procedure after link failure

*Case 1:* Link Failure can be Restored
In this case, $n_f$ detects the restoration signal of the failed link $e_f$. Then $n_f$ shifts the communication path from the cotree link of $L_r$ back to the restored link $e_f$ as seen in case 1 of Fig.3..

*Case 2:* Link Failure cannot be Restored
In this case, the structure of loops should be transformed in order to maintain a fundamental system of tie-sets. For example, the second network in Fig.2 does not maintain a fundamental system of tie-sets since $L_3$ contains two cotree links. To maintain a fundamental system of tie-sets, $n_f$ conducts the procedure which applies L-transformation [7] as shown in case 2 of Fig.3.

*3) L-transformation:* In this paper, L-transformation is defined as transformation of a fundamental system of tie-sets. If a formation of tree changes in a network, its fundamental system of tie-sets also changes in response to the transformed tree. Let $\boldsymbol{L}^f$ be a class of tie-sets that contains a failed link, and $L_r$ be a tie-set in which route switching is conducted. For each tie-set $L_i^f$ where $L_i^f \in \boldsymbol{L}^f \wedge L_i^f \neq L_r$, $n_f$ executes $L_i^f \leftarrow L_i^f \bigoplus L_r$. Then $n_r$ notifies the updated information by L-transformation to other nodes by means of advertisement based on tie-sets.

### B. Advertisement after L-transformation

After the procedure for link failure described in III-A, nodes around $n_f$ are still uninformed of the changes about updated communication paths and tie-sets. Therefore, state information of nodes relevant to link failure should be updated. A node relevant to link failure is defined as a node which belongs to a fundamental tie-set including the failed link $e_f$. State information can be updated by executing an advertisement based on massage passing on tie-sets. The massage passing is realized by sending *Update* massages around on tie-sets as shown in Fig.4. Time complexity of advertisement based on tie-sets is $O(D)$, where $D$ is a diameter of a graph. The

Fig. 4. Advertisement based on tie-sets after failure recovery

number of messages is equivalent to the number of tie-sets that contains a failed link. However, considering the worst case, communication complexity becomes $O(|E|)$.

## IV. Simulation and Experiments

A simulator is made by Java to verify the behavior of the recovery method for link failure suggested in this paper, and to compare against RSTP on behalf of existing technologies because of its general use. We did not conduct experiments on EAPS, since EAPS is not applicable to mesh topological networks such as a network shown in Fig.5. A tool which demonstrates RSTP is created using Delphi in reference to IEEE standards 802.1D [4]. In configuring a network, links are set to be undirected through which data can flow bi-directionally. In addition, network is designed to be redundant, in other words, bi-connected to be able to cope with failure as shown in Fig.5. As node configuration, each node has input ports and output ports, a message buffer, and a processor. Common buffering method is taken in a simulation node, where all messages received through input ports go to the message buffer. The processor takes each message from the message buffer by polling method. After each message is processed in the processor, the message is sent to other nodes through appropriate output ports unless it is received or discarded.

### A. Route Switching Points

The distinguished feature of the failure recovery based on tie-sets is that only one route switching is required to restore link failure. Generally, increase in route switching points leads to the factors as follows:

- Throughput degradation
- Slow recovery

Fig. 5. Network configuration consisting of 100 nodes created at random

For this reason, experiments to measure the number of times of switching required to restore one point link failure are conducted to compare against RSTP. A tree which represents communication paths before link failure is denoted as $T_o$, and a renewed tree which represents communication paths after link failure is denoted as $T_n$. To measure the number of route switching points, the distance between $T_o$ and $T_n$ is appropriate. The distance is defined as follows:

$$d(T_o, T_n) = |T_o - T_n| \tag{2}$$

Let $d_i(T_o, T_n)$ be the distance when link failure occurs on a tree link $e_i(\in T)$. Then the average of the number of route switching points $A_s$ is defined as follows:

$$A_s = \frac{\sum_{i=1}^{\rho} d_i(T_o, T_n)}{\rho}, \big(i = 1, 2, \ldots, \rho(= |T|)\big) \tag{3}$$

For a given bi-connected and undirected graph $G = (V, E)$, a graph $G$ is created at random with the number of nodes $|V|$ ranging from 20 to 100. A tree is output by giving link costs at random, and executing Spanning Tree Protocol (STP). Fig.6 is the experimental results that show the average $A_s$ of the number of route switching points. As shown in Fig.6, RSTP requires about twice as many switching as the proposed method on average, while recovery based on tie-sets needs only one time switching. In addition, $A_s$ shows modest upward tendency in a large-scale network.

Fig.7 shows the maximum times of route switching ($\max\{d_i(T_o, T_n)\}$) for each given graph whose condition is the same as the experiment to measure $A_s$. While route switching based on tie-sets constantly needs one shifting, RSTP requires much more switching than the proposed method. This is because RSTP greatly changes its tree topology in case of failure in the vicinity of a root bridge. Failure near a root node is the most major problem in operation of RSTP. The remarkable tendency of augmentation of the number of times of route switching is seen in a large-scale network. For example, 35 times of route switching, which can be seen in a graph with 100 nodes of Fig.7, greatly fluctuate the configuration of communication paths. In that case, throughput degrades seriously as well as convergence time greatly increases making an entire network unstable.

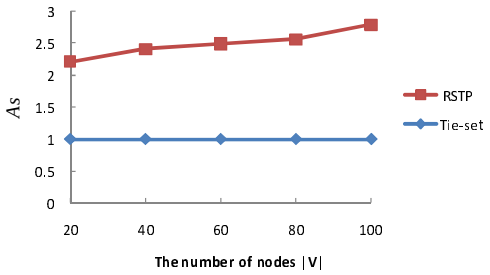


Fig. 6. The average times of route switching

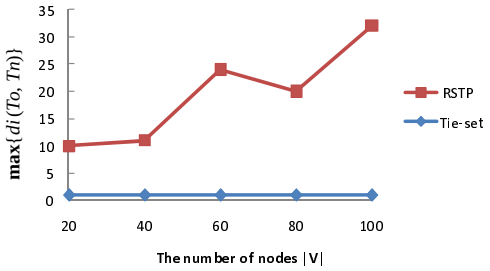


Fig. 7. The maximum times of route switching

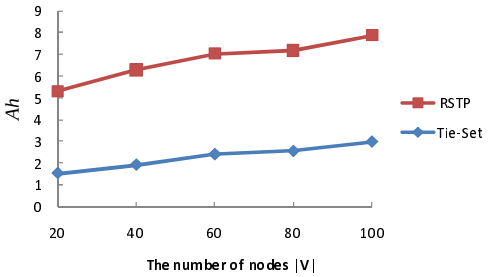


Fig. 8. The average number of hops

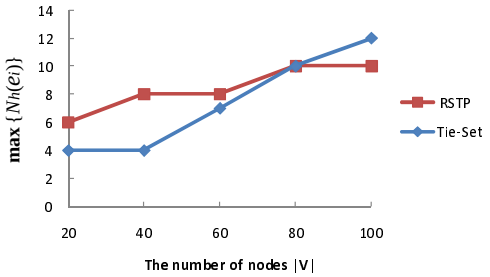


Fig. 9. The maximum number of hops

### B. Estimation of Recovery Time

In order to estimate recovery time for link failure, we counted the number of hops from a node that detects failure to a node that opens its physical port. In operation of RSTP, route switching often occurs several times. In that case, the most remote node from a root bridge should be counted since restoration finishes when the node sets its port state as *Destination Port*.

Let $N_h(e_i)$ be the number of hops from a failed point to a restored point when link failure occurs on a tree link $e_i(\in T)$. Then the average number of hops $A_h$ is defined as follows:

$$A_h = \frac{\sum_{i=1}^{\rho} N_h(e_i)}{\rho}, \big(i = 1, 2, \ldots, \rho(= |T|)\big) \qquad (4)$$

The conditions of network configurations are the same as the experiments on route switching points.

Fig.8 is the results that show the average number of hops $A_h$. As shown in Fig.8, recovery time of RSTP is about three times longer than that of the proposed method on average. Fig.9 shows the maximum number of hops ($\max\{N_h(e_i)\}$) for each given graph whose condition is the same as the experiment of $A_h$. As seen in Fig.9, the maximum number of hops of RSTP and the proposed method is almost the same. This is because the number of hops from a root node to an alternative link is almost the same in both methods when failure occurs in the vicinity of a root bridge. Therefore, as for recovery time, the proposed method realizes faster restoration on average than RSTP, although the worst case is balanced out.

### C. Influenced Nodes

Subsequently, the number of nodes which are influenced by link failure is counted. An influenced node is defined as follows:

a) *A node changing physical port states:* A state of communication paths on a network is determined by physical port states of network nodes. When a link failure occurs, it is necessary to open alternate ports to resume communication in addition to closing ports which are connected to the failed link. In the process of changing the states of communication ports, the of loss of frames occurs. The number of nodes that change their port states should be a criterion to measure reliability of a restoration method, since increase in influenced nodes in port states directly leads to instability of a network.

b) *A node changing state information:* State information of each node is updated by an advertisement. Until an advertisement is executed, a network stays unstable owning to discrepancy among state information of network nodes.

*1) Nodes that Change Physical Port States:* As mentioned, port states are important in data transfer. Therefore, if port states are changed by failure, the change of port states naturally influences communications on a network. Let $N_p(e_i)$ be the number of nodes that change their physical port states when link failure occurs on a tree link $e_i(\in T)$. Then the average number of nodes changing their port states $A_p$ is expressed as follows:

$$A_p = \frac{\sum_{i=1}^{\rho} N_p(e_i)}{\rho}, \big(i = 1, 2, \ldots, \rho(= |T|)\big) \qquad (5)$$

The conditions of network configurations are the same as the experiments on route switching points. Fig.10 is the results that show the average number of nodes which change their port states $A_p$. As shown in Fig.10, affected nodes in communications port states by RSTP are about twice as many as those by our method.

Fig.11 shows the maximum number of nodes that change

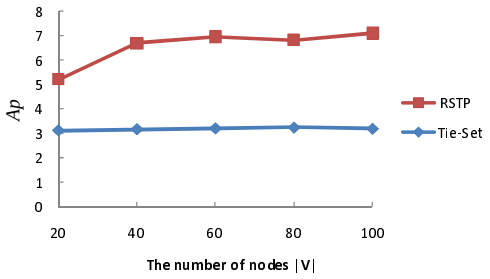


Fig. 10. The average number of nodes changing port states

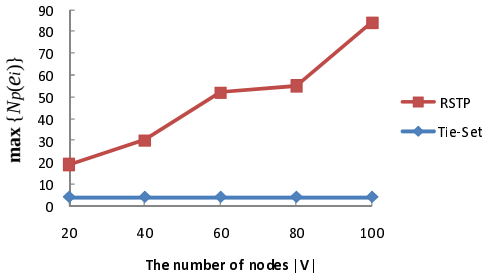


Fig. 11. The maximum number of nodes changing port states

their communications port states ($\max\{N_p(e_i)\}$) for each given graph whose condition is the same as the experiment to measure $A_p$. As seen in Fig.11, as a network scale becomes larger, the number of affected nodes in communications port states by RSTP greatly increase in the worst case. Thereby, communication reliability remarkably debases. The result shows a correlation between route switching points and influenced nodes. In other words, increase in route switching points also leads to low communication reliability.

*2) Nodes that Change State Information:* Failure recovery based on tie-sets executes update procedure of state information when there is a need for conducting an advertisement. The number of nodes influenced by an advertisement varies with tree structures. There are two major methods to output a tree; Breadth First Search (BFS) and Depth First Search (DFS). Focusing on the latter definition b) of influenced nodes, we conducted experiments to count the number of nodes that change their state information and to determine which tree is better. For a bi-connected undirected graph $G = (V, E)$ which is given at random, the range of the number of nodes $|V|$ is from 5 to 30. A tree is output by using BFS or DFS. In making a tree, a root node is set for a node that has the greatest number of incident links. Under these initial conditions, experiments were conducted to examine the scale influenced by an advertisement. Let $N_a(e_i)$ be the number of nodes that change their state information when link failure occurs on a tree link $e_i(\in T)$. Then the average number of nodes changing their port states $A_a$ is defined as follows:

$$A_a = \frac{\sum_{i=1}^{\rho} N_a(e_i)}{\rho}, \big(i = 1, 2, \ldots, \rho(= |T|)\big) \qquad (6)$$

Fig.12 is the results that show the average $A_a$. As shown in Fig.12, the BFS is more suitable than DFS since BFS can

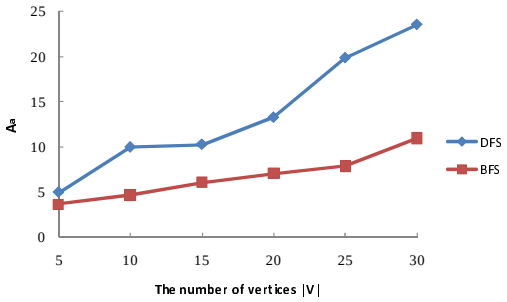


Fig. 12. The average number of nodes updating state information

reduce the number of nodes that change their state information in comparison with DFS.

## V. Conclusion and Future Work

In this paper, distributed control for link failure in information networks is suggested based on tie-set concept. As a result of experiments, we substantiate that the restoration based on tie-sets can reduce the scale affected by link failure in comparison with RSTP. Although one node has limited local information of tie-sets, an entire network is controlled in an orderly fashion due to the graph theoretical basis of tie-sets. Furthermore a series of local information of each node is consistent with the condition of an entire network.

As a future study, how to cope with concurrent link failure as well as node failure should be discussed. The proposed method easily works with RSTP, since our method employs the spanning tree algorithm. Therefore switch failure can be dealt with by adding some improvement of port states to the proposed protocol.

## References

[1] *Understanding SONET UPSRs*, http://www.sonet.com/EDU/upsr.htm.
[2] *Understanding SONET BLSRs*, http://www.sonet.com/EDU/blsr.htm.
[3] S.Shah, M.Yip, *Extreme Networks' Ethernet Automatic Protection Switching (EAPS) Version 1*, Network Working Group, Request for Comments: 3619, October 2003.
[4] IEEE Computer Society Sponsored by the LAN/MAN Standards Committee, *IEEE Standards 802.1D*, IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges, 9 June 2004.
[5] Michael Pustylnik, Mira Zafirovic-Vukotic, Roger Moore, RuggedCom, Inc., *Performance of the Rapid Spanning Tree Protocol in Ring Network Topology*
[6] N.Shinomiya, T.Koide, H.Watanabe, *A theory of tie-set graph and its application to information network management*, International Journal of Circuit Theory and Applications 2001; 29:367-379.
[7] T.Koide, H.Kubo, and H.Watanabe, *A study on the tie-set graph theory and network flow optimization problems*, International Journal of Circuit Theory and Applications 2004; 32:447-470.
[8] T.Koide, H.Watanabe, *A theory of Tie-Set Graph and Tie-Set Path - A Graph Theoretical Study on Robust Network System*, Proceedings of 2000 IEEE Asia Pacific Conference on Circuits and Systems; 1:227-230.
[9] K.Nakayama, N.Shinomiya, H.Watanabe, *Distributed Control for Link Failure Based on Tie-Sets in Information Networks*, Proceedings of 2010 IEEE International Symposium on Circuits and Systems; pp.3913-3916.
[10] Nancy A. Lynch, *Distributed Algorithms*, Morgan Kaaufmann Publishers, Inc.: San Francisco, California, 1996.
[11] Iri M, Shirakawa I, Kajitani Y, Shinoda S etc., *Graph Theory with Exercises*, CORONA Pub: Japan, 1983.
[12] Swamy MNS, Thulasiraman K., *Graphs, Networks, and Algorithms*, Wiley Interscience: New York, 1981.