A System for Constructing Spanning Trees in Graph Networks that Utilize Integer Linear Programming to Enhance Link Fault Tolerances

Hieu Tran The, Kosuke Fujita, and Nattapong Kitsuwan

Department of Computer and Network Engineering
The University of Electro-Communications, Tokyo, Japan

***Abstract*—In this paper, we formulate to optimize the problem related to network failure using the integer linear programming (ILP) method. We aim to minimize the number of spanning trees needed to protect the network in case of link and/or link-node failure. Compared to the traditional approach of constructing spanning trees using heuristic algorithms, our method has successfully reduced the number of trees required for node or link-node failure protection by up to 50%. Reducing the number of spanning trees saves memory requirements for the network route and simplifies network configuration.**

***Index Terms*—Network protection, integer linear programming, spanning trees, link failure, reducing memory.**

I. INTRODUCTION

In the contemporary era of highly interconnected networks, failures can have significant repercussions for both individuals and organizations. Consequently, it is imperative to establish robust and resilient networks capable of withstanding a wide range of failures. Numerous techniques have been proposed to promptly detect and mitigate network failures as they arise. One such technique is the open shortest path first (OSPF) protocol [7], which relies on link-state routing. OSPF can identify link failures, acquire knowledge of network topology, and adapt to new topologies. However, there are instances where OSPF's recovery time may exceed one second or even more, rendering it potentially inadequate for certain applications.

An alternative approach is P4Resilience [8], which combines the principles of software-defined networking (SDN) with programmable data planes, utilizing P4. P4Resilience employs packet headers to encapsulate backup path information, streamlining failure recovery. It also implements a loop-free backup path algorithm that enhances resilience against multi-link failures. Nonetheless, it is important to note that the addition of headers to packets could potentially lead to a reduction in network performance, and the deployment of P4 switches might result in increased network costs.

Moreover, there exists a solution that overcomes the disadvantages of the above-mentioned options. This solution has a fast recovery time in case of link failure and eliminates the requirement of supplementary hardware devices and packet headers, and it is called a multi-VLAN-based approach. In the multi-VLAN-based approaches, failure recovery is achieved by switching the traffic affected by a failure to a backup VLAN tree to bypass the failure device. The backup VLANs are preconfigured from the spanning tree and stored in each switch. The failover time is significantly reduced compared to a traditional ethernet network because each switch performs failure recovery based on local decisions [6]. Figure 1 depicts an example of VLAN based protection scheme. In this example, VLAN 1 is the working VLAN, and VLAN 2 is used to protect link $(5,8)$. In the normal state, node 8 uses link $(5,8)$ to deliver frames to the destination nodes $1,2$, and 3. As link $(5,8)$ fails, node 8 uses VLAN 2 to send frames to avoid using the failure link.

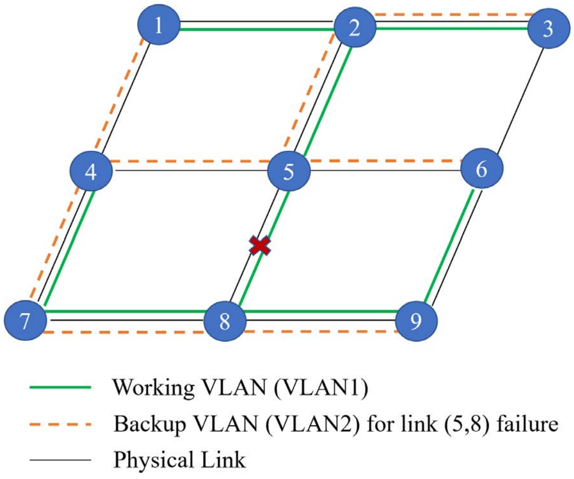


Fig. 1. Example of VLAN-based protection scheme

We need to build a set of spanning trees to use a multi-VLAN-based approach. Traditional methods for constructing spanning trees use heuristic algorithms, such as the one presented in [4]. However, these methods may not be optimal regarding the number of spanning trees and implementing them constrained nodes and links in a specific order, which can lead to increased memory consumption and routing complexity.

In this paper, we formulate the failure problem in the network as an integer linear programming (ILP). Specifically,

---

Identify applicable funding agency here. If none, delete this.

our goal is to minimize the number of spanning trees needed to protect the network in the event of link or/and node failure. Our method has successfully reduced the number of spanning trees required for nodes or link-node failure by up to 50% in JPN25 [1] and NSFNet [9] network while maintaining protection against link or/and node failure. This reduction not only reduces memory requirements for network routing but also simplifies the network and makes it more manageable.

II. PROBLEM STATEMENT

Spanning trees are often used to protect against link and/or node failures in distributed networks, as described in [2], [3], or centralized networks [5]. The number of potential spanning trees that can be established in the network to ensure network protection ranges from $2 \rightarrow n^{(n-2)}$, where n represents the number of nodes in the network. In the case of a large network, constructing a set of spanning can result in an excessive number of trees, which can lead to increased memory usage for VLAN storage, adding complexity and cost charges to the network configuration. Therefore, our goal is to minimize the number of spanning trees needed to be created while maintaining protection against any possible link or/and node failures in the network.

III. ILP FOR THE FORMULA TO OPTIMIZE THE PROBLEM

To use ILP for the formula to optimize the problem, the parameters are defined in the table (I):

TABLE I
DESCRIPTION OF PARAMETERS AND VARIABLES

V	represents the set of nodes (routers or switches) in the network
E	represents the set of links in the network
T	represents the set of spanning trees to construct
r	root node in spanning tree $r \in V$
M	large enough value
ε	small enough value
k^t	equal 1 if there is a spanning tree $t \in T$, 0 otherwise
x_{ij}^t	number of lower node under link $(i, j) \in E$ in spanning tree $t \in T$, $1 \leq x_{ij}^t \leq \lvert V\rvert - 1$
y_{ij}^t	equal to 1 if link $(i, j) \in E$ of the spanning tree $t \in T$ exists, 0 otherwise
w_i^t	equal to 1 if node $i \in V$ is the leaf of the spanning tree t, 0 otherwise
z_i^t	degree of node $i \in V$ in tree $t \in T$, $1 \leq z_i^t \leq \lvert V\rvert - 1$ $(i \in V)$

The objective function is to minimize the number of spanning trees and is defined as Eq. (1) as follows:

$$\min \sum_{t \in T} k^t \tag{1}$$

A. Link failure

Link failure constraint to enhance link fault tolerance. The link failure constraints require that every link in the graph belongs to at least one of the set of spanning trees. If a spanning tree satisfies a link constraint, and more than one tree does so, any link failures on the graph will be covered by at least one of the spanning trees. First, we apply constraints for the number of links in spanning tree conditions.

$$\sum_{(i,j) \in E} y_{ij}^t = k^t \, (|V| - 1), \quad \forall \, t \in T \tag{2}$$

Equation (2) expresses that the sum of links in the spanning tree is $|V| - 1$. Next, we consider the following conditions to describe the flow conditions in the network.

$$\sum_{(i,j) \in E} x_{ij}^t \leq k^t \, M, \quad \forall \, t \in T \tag{3}$$

Equation (3) limits the total flow through tree $t \in T$ by a large constant number M. Next, we show the constraints on which links are in the spanning tree.

$$x_{ij}^t \leq y_{ij}^t \, (|V| - 1), \quad \forall (i, j) \in E, \, t \in T \tag{4}$$

Equation (4) these constraints for removing the loop in the spanning tree and ensures that if link (i, j) is included in tree $t \in T$, the number of a lower node of each link in the spanning tree is always less than $|V| - 1$. The following are two formulas Eqs. (5) and (6). Constraints of flow between node root and any nodes in the network. More detailed description, for any link pairs (r, i) of node $r, i \in V$ where a path exists between r and i with direction from r to i.

$$\sum_{(r,j) \in E} x_{rj}^t - \sum_{(j,r) \in E} x_{jr}^t = (|V| - 1)k^t, \quad \forall t \in T \tag{5}$$

$$\sum_{(i,j) \in E} x_{ij}^t - \sum_{(j,i) \in E} x_{ji}^t = -k^t, \quad \forall t \in T, \forall i \in V, i \neq r \tag{6}$$

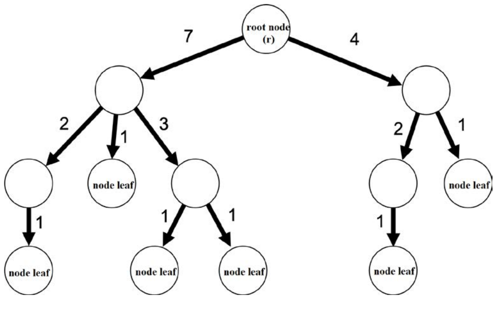


Fig. 2. Convert of flow rate from the root node in a spanning tree

Equation (5) ensures a flow between root node r and any node j. Equation (6) is flow constraints for flow balance in all of the nodes except for the root node. For example, in Fig. 2. A directed graph consists of $V = 12$ nodes and the values of x are shown. We need $(|V| - 1) = 11$ flows to connect all the nodes from the root node r. In addition, for non-root nodes, the total number of flows leaving the node is always less than the total number of flows entering a flow. The next equation (7) ensures link disjoint in the spanning tree.

$$\sum_{t \in T} (y_{ij}^t + y_{ji}^t) \leq \sum_{t \in T} k^t - 1, \quad \forall (i, j) \in E \tag{7}$$

Equations (1) to (7) are presented above to minimize the spanning tree in case of link failures.

B. Node failure

Next, we will build a spanning tree in case of node failure. Constraints in eqs. (1) to (6) are the same as link failure constructions, and we are not use constraint link disjoint (7) in this case. We will explain the newly added constructions to protect against node failure. First, we present constraints for nodes as leaves in the spanning tree, in Eq. (8).

$$\sum_{i\in V} w_i^t \le k^t\,|V| - 1, \quad \forall\, t \in T \tag{8}$$

Next, we show the equation to calculate the degree of each node in the spanning tree.

$$z_i^t = \sum_{(i,j)\in E} y_{ij}^t + \sum_{(j,i)\in E} y_{ji}^t, \quad \forall t \in T, \forall i \in V \tag{9}$$

Following, we present constraints to satisfy the node conditions.

$$(1+\varepsilon)k^t - z_j^t - M w_j^t \le 0, \forall\, j \in V,\ \forall\, t \in T \tag{10}$$

$$z_j^t - 1 - M(1 - w_j^t) \le 0, \quad \forall\, j \in V, \forall\, t \in T \tag{11}$$

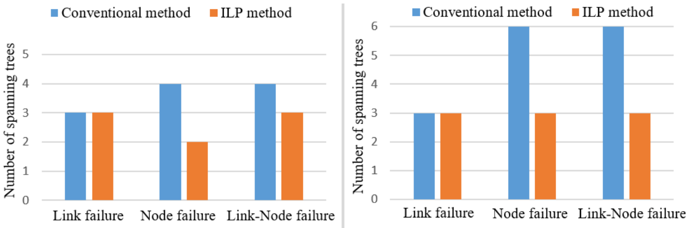
Equations (10) and (11) determine whether node $j \in V$ in the spanning tree $t \in T$ is a leaf. Where ε is a sufficiently small positive real number, and M is a sufficiently large positive real number. The fact that if node $j \in V$ is a leaf means that the degree of node j is $z_j^t = 1$, then $w_j^t = 1$. On the other hand, if the degree of the node is greater than 1, then $w_j^t = 0$.

C. Link-node failure

It can be guaranteed that network problems in case the link or the node fails. Applying the constraint conditions required for the node constraint failure combined with the constraint in Eq. (7).

IV. PERFORMANCE ANALYSIS

This section compares our formulated ILP and the heuristic algorithm-based approach in [4]. To conduct the comparison, we evaluate the performance of both methods on two networks: the Japan Photonic Network (JPN25), and the National Science Foundation Network (NSFNet). The evaluation results are presented in Fig. 3, showing to compare the number of spanning trees required for network protection against link and/or node failure. From the results, it is observed that our proposed ILP method outperforms the traditional heuristic algorithm-based approach regarding the number of spanning trees required to protect against node and link-node failures. In the NSFNet network, the number of spanning trees required node and link-node failures protection is 2 and 3, respectively, while the traditional approach is both 4, corresponding to a reduction of 50 % and 25 %. In the second network evaluated as JPN25, we have reduced the number of spanning trees required by 50 % for node and link-node failure protection.



(a) NSFNet **(b) JPN25**

Fig. 3. Number of spanning trees for failure condition.

By reducing the number of required spanning trees for network protection, our proposed method also reduces the memory demand for storing VLAN configurations, which are used in network routing. This makes our method more efficient in optimizing network design in the event of failure.

CONCLUSION

In this paper, we formulated the link and/or node failure problems as an integer linear programming (ILP) problem to minimize the number of spanning trees necessary for protection against link and/or node failure in the network. Compared to the traditional approach of constructing spanning trees using a heuristic algorithm, our method has successfully reduced the number of required spanning trees that need to be built by 50 %, thereby reducing the memory demand for network routing and decreasing network configuration complexity. Moreover, This work provides the foundation for future research to develop memory-optimized and efficient methods to protect networks from link and/or node failures.

REFERENCES

[1] "Topical characteristics of the Japanese photonic network model, "IEICE Technical Report, PN2013-2, June 2013

[2] B. Zhou et al. "HDEER: A Distributed Routing Scheme for Energy-Efficient Net-working" IEEE Journal on Selected Areas in Communications, vol. 34, no. 5, pp. 1713-1727, May 2016.

[3] "DRAFT IEEE Standard for Local and metropolitan area networks-Virtual Bridged Local Area Networks" in Proc. IEEE Unapproved Draft Std P802.1Q, 2009 Edition/D0.1, Feb. 2010.

[4] J. Farkaset al. "Distributed resilient architecture for Ethernet networks" in Proc. 5th International Workshop on Design of Reliable Communication Networks, 2005.

[5] S. Sharma et al. "Viking: a multi-spanning tree Ethernet architecture for the metropolitan area and cluster networks" in Proc. IEEE INFOCOM 2004, Hong Kong, China, 2004.

[6] Kitsuwan et al. "Independent Transient Plane Design for Protection in OpenFlow-Based Networks" Journal of Optical Communications and Networking, vol. 7, no. 4, Apr. 2015.

[7] Goyal et al. "Achieving faster failure detection in OSPF networks" in Proc. IEEE Int. Conf. on Communications (IEEE ICC), USA, 2003.

[8] Li Ziyong et al. "P4Resilience: Scalable resilience for multi-failure recovery in SDN with the programmable data plane" The International Journal of Computer and Telecommunications Networking, Vol. 208, 2022.

[9] Rekhter et al. "EGP and Policy Based Routing in the New NSFNET Backbone", T.J. Watson Research Center, IBM Corporation, March 1988. Also, as RFC 1092, February 1989.