Reducing Link Failure Test Cases for Telecom Networks by Focusing on Topological Similarity

Megumi Shibuya[†], Toshihiko Kato[†], Teruyuki Hasegawa[††] and Hirozumi Yamaguchi[‡]
[†]University of Electro-Communications
Tokyo, Japan
e-mail: shibuya@net.is.uec.ac.jp
kato@is.uec.ac.jp
[††]KDDI, Corp.
Tokyo, Japan
e-mail: teruyuki0077.society@gmail.com
[‡]Osaka University
Osaka, Japan
e-mail: h-yamagu@ist.osaka-u.ac.jp

***Abstract*—In order to provide high reliable network services for a large number of users, an ideal network test is required to guarantee end-to-end connection for any pair of clients with a failure of any link. However, the time and human resources that can be dedicated to the test are limited, which prevents such an ideal test. In this paper, we propose a method to reduce the number of network failure test cases by focusing on "similarity" among test cases. We leverage topological similarity for the purpose, formulate a problem of minimizing the number of test cases, and conduct simulation experiments to show the effectiveness.**

Keywords- Telecom network; Link failure test; Topological similarity; Optimization.

I. INTRODUCTION

The telecom companies, such as Internet service providers and mobile network operators, are required to provide highly reliable network service for a large number of users. For this purpose, the networks maintained by telecom companies, which we call *telecom networks* in this paper, take the following two approaches. The first is to introduce redundant configuration. At least two paths are guaranteed between any two network node pair. The second is strict testing at changing network configuration, such as introducing new equipment, changing the network configuration, and upgrading software versions.

In general, the testing in telecom networks is conducted as link failure test. In the case that a physical link is disconnected, or network equipment, such as a switch and a router is broken, those failures are detected as failures at the related links. Therefore, it will be common that the testing at the network configuration change is performed in the following way.

- Select a pair of nodes which may communicate together, and determine a communicating path between them.
- Pick up one of the links comprising the path, cause fault artificially on the selected link (*link shutdown*), and verify that a new communication path is set up and the communication between the selected nodes is restarted. This procedure is called a *test case* in this paper.
- Conduct the above test case for all possible node pairs and links in paths.

Recent telecom networks have become large in their size and complex in their structure. This increases the number of test cases in the link failure testing. Furthermore, due to the spread of Internet of Things (IoT) applications, the number of devices connected to network is increased, the number of communication node pairs is increased, and the number of links included in a path between a node pair is increased. As a result, the number of link failure test cases is drastically increased. On the other hand, it is difficult for a telecom company to conduct a vast number of tests exhaustively within a limited time until the start of service. Therefore, it becomes important to reduce the number of failure test cases without deteriorating the reliability.

We focus on the topological similarity in a telecom network to reduce the number of test cases. In general, a telecom network has a tree-like hierarchical structure, consisting of several *link tiers* including core, relay and access. The type of network equipment and the bandwidth of links are selected on a tier basis, and the type of routing protocols also depends on tiers, e.g., border gateway protocol (BGP) at a core tier and open shortest path first (OSPF) at a relay tier.

Therefore, we propose to introduce the topological similarity to selecting the link failure test cases in the following two steps.

Step 1:

We introduce the *similarity of test cases* using an idea of the *abstract path* that considers only the number of links and their tiers in a path.

Specifically, for different communicating node pairs whose paths have the same number of links, each of which has the same link tier (the same abstract path), we suppose that the communications for these node pairs will be processed similarly within the telecom network. Based on this concept, we define that test cases for node pairs and shut down links are *similar*, when the followings are satisfied.

(1) The paths for node pairs before a link shutdown have the same abstract path.
(2) The shut down links are located at the same tier.
(3) The paths for node pairs after a link shutdown have the same abstract path.

For reducing the number of test cases, we suppose that conducting only one of test case will be enough among the similar test cases.

Step 2:

We expand the idea of abstract path so as to include other attributes.

In Step 1, only the number of links and the tier of links are considered. However, other attributes, such as the link bandwidth and the manufacturers or types of network equipment will influence the processing behaviors within the network. So, we add link bandwidth or equipment type in the definition of abstract path.

In this paper, we propose how to reduce the number of link failure test cases by focusing on the topology similarity based on the idea of test case similarity using abstract path. We suppose an IoT oriented telecom network, where end nodes in the network, which we call *clients*, communicate with each other, and the link failure test assuming single link failure. We define as a *test target* a set of communicating client pair and a shut down link over the path between the client pair. We cluster the possible test targets throughout a telecom network into groups of similar test targets based on the idea of abstract path, as described in Steps 1 and 2 above. This will be done by formulating with integer linear programming (ILP) problem. By selecting one test target from a test target cluster, we can reduce the number of test cases drastically.

This paper is organized as follows. Section II summarizes related work and Section III explains the proposed method. Section IV presents the experimental results of applying the proposed method to various types of networks. We conclude this work in Section V.

II. RELATED WORK

As for the network failure test, there are many studies focusing on improving its efficiency by automating network tests [1]-[3]. M. Shibuya et al. [1], our previous paper, proposes an automated network testing system which can automatically construct test environment, check connectivity, test scenario execution, and collect and analyze test results. The work in [2] proposes L1 patch where all OpenFlow switches are virtually monitored in a single L1 patch panel using an OpenFlow technique. By combining this approach with Mininet [4], an OSS-based network testing tool mitigates the operational workload. The dedicated time is just a few minutes per operator, while the conventional work procedure requires a half hour or an hour by two operators. The work in [2] also reports that the number of test types increased to 194, which had been just 90 in the conventional approach because the testing time was shorten. The work in [3] proposes a method to construct an automated test platform for Virtual Network Functions (VNF). Network comprehensive tests, which had required a lot of manual operations so far, can be automatically performed by using the proposed tool [5]. According to [3], 2,736 types of tests can be conducted in 40 hours. These workload mitigation methods are of great help, but we need to design a fundamental methodology to reduce the number of test cases while keeping the test coverage.

As for the topology similarity, there are several studies on the graph similarity based on network metrics, such as hop distance and link degrees [6]-[10]. The graph isomorphism problem [6] is a computational approach to determine whether two finite graphs are isomorphic or not. This approach is able to prove the isomorphism between two graphs, and some algorithms, such as Ullman [7], VF 2 [8], and Naughty [9], are proposed. However, they have an issue that the computational complexity is not known.

G. Kollias et al. [10] proposes a similarity calculation method of topology in a neuro network. This method compares the similarity of partial networks focusing on the number of connections of nodes and links. For network tests, however, we need to take into account of not only the topology but also the end user paring, routing and link failures. Therefore, the challenge is to design an appropriate similarity that actually covers a wide variety of viewpoint.

III. PROPOSED METHOD

In this section, we explain how our proposed method reduces the number of link failure test cases by focusing on the topology similarity based on the test case similarity using abstract path.

A. Traditional Failure Test

As described above, a traditional failure test requires that testing be conducted for all combinations of client pairs and shut down links over paths between individual client pairs. Therefore, the number of test cases in a traditional test is given by the following;

$$F = \sum_{c_i, c_j \in C, i<j} H_{ij} \tag{1}$$

where C is the set of clients and H_{ij} is the path length between clients c_i and c_j.

B. Notation

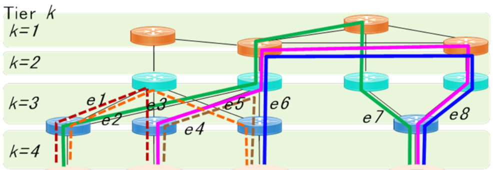
As described in Section I, the purpose of proposed method is to reduce the number of link failure test cases by introducing similarity among test cases in two steps. In order to specify the proposed method, this paper introduces the following notations.

- a path between clients c_i and c_j: *Primary* (c_i, c_j)
- test target focusing on clients c_i and c_j, and link to be shut down (shut down link) e: (c_i, c_j, e)
  Here, $e \in$ *Primary* (c_i, c_j).
- a path between clients c_i and c_j after shut down link e is shut down: *Secondary* (c_i, c_j, e)
- path length of a client pair (c_i, c_j) before link shutdown: q
- path length of a client pair (c_i, c_j) after link shutdown: r
- the link tier of shut down link e: k
- test target attribute vector used in Step 1: (q, r, k)

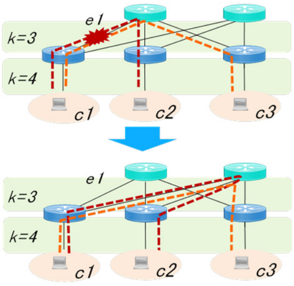
- name of additional test attribute used in Step 2: α
  In this paper, link bandwidth and equipment type are used as α.
- value of attribute defined dependent of attribute name α: w_α
  In this paper, w is defined as follows;
  - α = bandwidth: w_α = link bandwidth with shut down link *e*
  - α = equipment_type: w_α = equipment types at both sides of shut down link *e*
- test target attribute vector used in Step 2: (q, r, k, w_α)
- maximum number of conducting test cases: N

C. Basic Idea

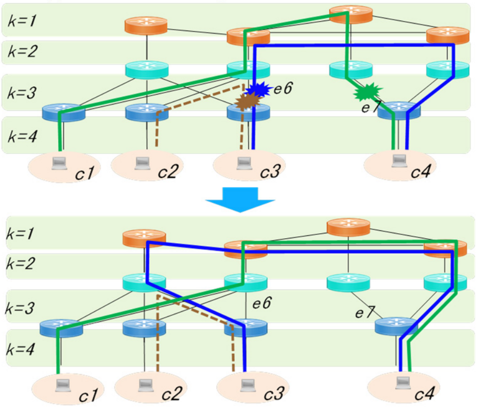
In this subsection, we describe our basic idea using an example shown in Figure 1. Here, we suppose four link tier network with four clients. Figure 1(a) shows the paths between possible client pairs. We suppose the shutdown of links at tier 3 (links *e1* though *e8*). Figure 1(b) shows a test case where link *e1* is shut down. In this case, Paths *Primary* (*c1, c2*) and *Primary* (*c1, c3*) will be rearranged to paths *Secondary* (*c1, c2, e1*) and *Secondary* (*c1, c3, e1*), respectively, as shown in the figure. Figure 1(c) shows the test cases where link *e6* or *e7* is shut down. When we focus only on the path lengths before and after link shutdown (*q* and *r*, respectively), and the tier of shut down link (*k*), the paths can be modeled as shown in Figure 2. This is abstract paths handled in Step 1. For all the paths among clients *c1*, *c2,* and *c3*, the paths before and after link shutdown have path length 4, i.e., $q = r = 4$. In this case, we consider that all the test cases whose test targets are specified by (i) in Figure 2 are similar and can be clustered in a group (test target cluster). We propose that one test case can represent this cluster and that checking one test case is enough in the link failure test.



(a) Network configuration and paths between client pairs.



(b) Test case focusing on shutdown at link *e1*.



(c) Test case focusing on shutdown at link *e6* or *e7*.
Figure 1. Example of link failure test.

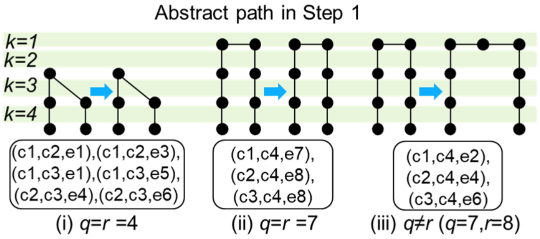


Figure 2. Test target clustering in Step 1.

For the paths between *c1*, *c2* or *c3*, and *c4*, there are two cases as shown in (ii) and (iii) in Figure 2. The path length before link shutdown, the path length is 7 for all test targets. For test targets (*c1, c4, e7*), (*c2, c4, e8*) and (*c3, c4, e8*), the path length after link shutdown is 7. Figure 1(c) shows the case of test target (*c1, c4, e7*). On the other hand, for test targets (*c1, c4, e2*), (*c2, c4, e4*) and (*c3, c4, e6*), the path length after link shutdown is 8. So, we consider that the behaviors after the link shutdown will be different for these two test case clusters, and we propose that one test target needs to be selected from each of the clusters.

As we explained in Section I, the abstract path in Step 1 is too generalized network features. So, in this paper, we add other attributes, such as link bandwidth and equipment type, in defining abstract paths. Consider that, among links at tier 3, link *e5* has different bandwidth than the others. Say, the bandwidth of link *e5* is 1 and that of others is 2. In this case, the test target cluster given by (iii) in Figure 2 is divided as shown in Figure 3. Specifically, only in the test case for target (*c3, c4, e6*), the path after link shutdown goes through link *e5*, as indicated in Figure 1(c). Therefore, test targets (*c1, c4, e2*) and (*c2, c4, e4*) are classified in a cluster and target (*c3, c4, e6*) is in another.

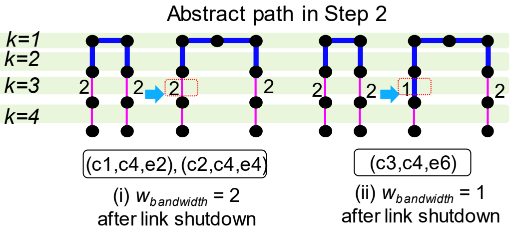


Figure 3. Test target clustering in Step 2.

ISBN: 978-1-61208-650-7

D. Formulation

In this subsection, we formulate how to group test targets into test target clusters as an optimization problem.

Suppose a network that consists of switches $V = \{v_1, v_2, \ldots, v_S\}$ and links $E = \{e_1, e_2, \ldots, e_L\}$, whose sizes are respectively S and L and follows k tiered topology. It is expressed as network graph $G = (V, E)$ with $E \subseteq V \times V$. Each client is a member of the client set $C = \{c_1, c_2, \ldots, c_M\}$ *where* M denotes the number of clients, and is connected to one of switches as a leaf. Each link has an additional attribute and has an attribute value assigned from $W = \{w_{e1}, w_{e2}, \ldots, w_{eL}\}$.

Next, we define the similarity function *Sim ()* between two test targets in the following way. That is, if the test target attribute vectors are the same for two test targets, then this function is 1, otherwise 0.

$$Sim\ \left(\left(c_x, c_y, e\right), \left(c_a, c_b, e_z\right)\right) = \begin{cases} 1 & same \\ 0 & otherwise \end{cases} \qquad (2)$$

Using on this function, the set of similar test targets are calculated in the following way, focusing on a specific link e to be shut down, which we denote as $D(e)$.

$$\begin{aligned} D(e) = &\left\{\left(c_x, c_y, e\right) \mid \forall c_x, \forall c_y \in C;\ x < y\right\} \\ &\cup\ \left\{\left(c_a, c_b, e_z\right) \mid \forall c_a, \forall c_b \in C;\right. \\ &\quad \forall e_z \in E; \exists c_i, \exists c_j \in C; \\ &\quad i < j;\ a < b; \\ &\quad \left. Sim\ \left(\left(c_i, c_j, e\right), \left(c_a, c_b, e_z\right)\right) = 1\right\} \end{aligned} \qquad (3)$$

Furthermore, we introduce the binary variable *down*(e) that indicates whether the link shutdown at link e is actually conducted (=1) or not (=0). $Z(e)$, a subset of $D(e)$ that only includes test targets associated with actual shutdown links, can be given with *down*(e) as follows.

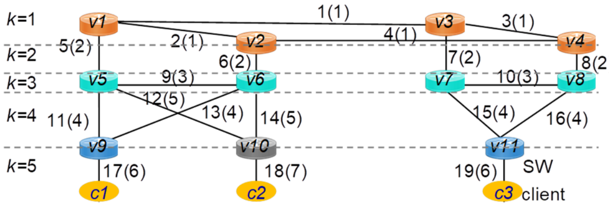
$$Z(e) = \begin{cases} D(e) & if\ down(e) = 1 \\ \varphi & if\ down(e) = 0 \end{cases} \qquad (4)$$

Here, the number of shut down links must not be greater than N, which is a given limit.

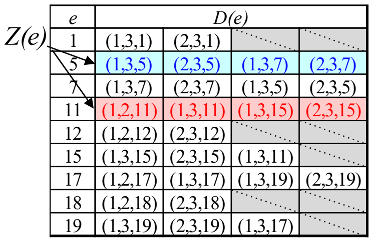
$$\text{s.t.} \qquad \sum_{e \in E} down(e) \le N \qquad (5)$$

By selecting $Z(e)$ with the maximum number of test targets, the number of test cases is reduced. Therefore, using (2) through (5), the objective function to be optimized is expressed in (6).

$$\text{maximize} \qquad \sum_{e \in E} \left| \cup Z(e) \right| \qquad (6)$$



(a) Example network



e	D(e)			
1	(1,3,1)	(2,3,1)		
5	(1,3,5)	(2,3,5)	(1,3,7)	(2,3,7)
7	(1,3,7)	(2,3,7)	(1,3,5)	(2,3,5)
11	(1,2,11)	(1,3,11)	(1,3,15)	(2,3,15)
12	(1,2,12)	(2,3,12)		
15	(1,3,15)	(2,3,15)	(1,3,11)	
17	(1,2,17)	(1,3,17)	(1,3,19)	(2,3,19)
18	(1,2,18)	(2,3,18)		
19	(1,3,19)	(2,3,19)	(1,3,17)	

(b) Similar test targets for example network

Figure 4. Example of test target similarity.

Figure 4 shows an example of calculating similar test targets. Figure 4(a) gives an example network with 5 tiers. Each link is associated with its name and an attribute value in the form of name (attribute). For example, test targets (1,3,11) and (2,3,15) in Figure 4 are similar, i.e., *Sim*((1,3,11), (2,3,15)) = 1, because q, r, k and w of both paths are the same (q=7, r=7, k=4 and w=4).

As the result of applying (2) and (3), the similar test targets are calculated for this network as shown in Figure 4(b). After that, by considering (4) and (6), the optimal test targets are selected. If N = 2, $Z(5)$ and $Z(11)$ are selected as shown in Figure 4(b).

Since this can be treated as an ILP problem, we can calculate the optimal solution by the general programing solver, e.g., IBM ILOG CPLEX.

IV. EXPERIMENTAL EVALUATION

We present the experimental results of applying the proposed method to various types of networks in the following subsections.

A. Experimental Setup

In order to evaluate the effectiveness of our proposed method, we have set up the solver IBM ILOG CPLEX Optimization Studio12.6 [11] on Windows PC (OS: Windows 8.1 Pro, CPU: Intel Core (TM) i7-6700@3.40GHz, memory: 16.0GBytes), and calculate the optimal set of shut down links.

As shown in Table I, we have applied 6 topologies, where the numbers of switches, clients, links and link tiers are different each other. Four network topologies (NW1 though NW4) are based on ladder type with path redundancy. The last two topologies (NW5 and NW6) are based on the core network topologies of Japanese telecom company [12] and Japanese internet project [13], by adding relay and access tiers with redundant topologies. The network topology of NW6 is shown in Figure 5.

TABLE I. EXPERIMENTAL NETWORK TOPOLOGIES

NW #	total nodes (#SWs+#Clients)	#links	#link tier *k*	max. path length *q*	#band-width	#equipment types
1	14 (11+3)	19	4	7	6	14
2	16 (12+4)	22	4	7	6	12
3	38 (30+8)	54	4	9	6	16
4	72 (56+16)	106	4	13	6	16
5	100 (74+26)	138	6	11	7	29
6	232 (174+58)	346	4	13	11	16

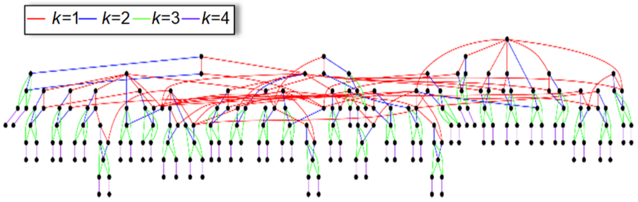


Figure 5. Example of evaluation network topology (NW6).

The paths before and after link shutdown are calculated by the Dijkstra's algorithm. First, the shortest path of each client pair is calculated as Primary (). Next, by removing a shut down link from the primary path, Secondary () is calculated. The similarities between any pair of test targets are calculated according to the elements of the test target attribute vector *q, r, k and w* described in Section III-B.

TABLE II. EVALUATION CASES

Case#	Evaluation Methods	Method
1	Comprehensive test for all test targets (=*F*)	Traditional
2	Test target attribute vector (q, r, k)	Proposed
3	Test target attribute vector ($q, r, k, w_{bandwidth}$)	
4	Test target attribute vector ($q, r, k, w_{equipment\_type}$)	

To evaluate the effectiveness of proposed method, we compared the four cases as shown in TABLE II. Case1 is a traditional method, and the number of comprehensive test for all test targets is *F* calculated by (1). Case2, Case3 and Case4 are proposed method, and each their test target attribute vectors are different. Note that the links not comprised in the path between any client pairs are exclude from the calculation.

B. Test Case Count and Reduction Ratio

First, we evaluate the test case count and the reduction ratio compared with comprehensive test for all test targets.

The reduction ratio *E1* is calculated by (7). The number of test cases and the reduction ratio are shown in TABLE III.

$$E1 = 1 - \frac{N\ in\ CaseX}{F} \quad \left(X = 2,3,4\right) \qquad (7)$$

From the table, the number of test cases of the proposed methods of Case2, Case3 and Case4 in each topology were massively reduced compared with the traditional method. In almost all cases, the reduction ratio *E1* was decreased in order of Case2, Case3 and Case4. Specifically, the reduction ratio *E1* of Case4 at NW1, and Case2 at NW6 were 0.833 (minimum) and 0.999 (maximum), respectively.

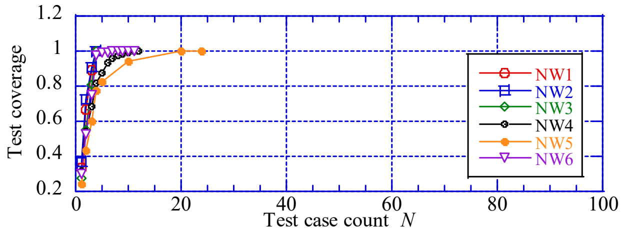
TABLE III. NUMBER OF TEST CASES AND REDUCTION RATIO

NW #	Test case count (reduction ratio)			
	Case#			
	1	2	3	4
1	36 (0)	4 (0.889)	5 (0.861)	6 (0.833)
2	66 (0)	4 (0.939)	6 (0.909)	7 (0.894)
3	404 (0)	4 (0.990)	7 (0.983)	10 (0.975)
4	2120 (0)	4 (0.998)	19 (0.991)	36 (0.983)
5	5424 (0)	19 (0.996)	19 (0.996)	53 (0.990)
6	29896 (0)	11 (0.999)	91 (0.997)	69 (0.998)

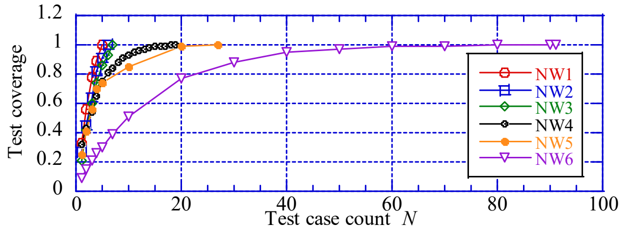
From the above results, it was confirmed that our proposed method is able to reduce the test case count significantly even when the test target attribute is considered.

C. Test Case Count and Test Coverage

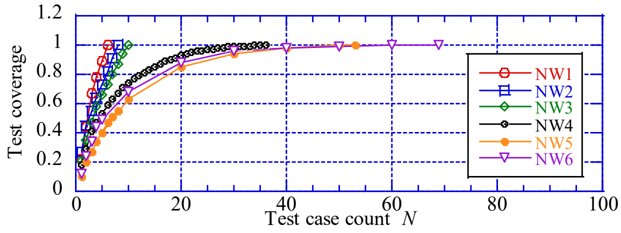
Next, in order to complete the link failure tests within a limited time, the number of link failure test cases may be limited. Therefore, we evaluate the test case count *N* and the test coverage. The test coverage is defined as the ratio of the number of similar test targets at the test case count *N* against *F*.



(i) Test target attribute vector (q, r, k) (Case2).



(ii) Test target attribute vector ($q, r, k, w_{bandwidth}$) (Case3).



(iii) Test target attribute vector ($q, r, k, w_{equipment\_type}$) (Case4).

Figure 6. Test case count *N* vs test coverage.

Figure 6 shows the test coverage in response to different test case count (*N*), in each evaluation case. From this figure, the test coverage increased in response to the number of test case count N in all cases. In addition, the small-scale network, such as NW1, NW2 and NW3, the test coverage was increased rapidly. The test coverage increased in response to N rapidly in order of Case2, Case3, and Case4, except at NW6. In Case4 at NW6 in Figure 6 (iii), the test coverage was almost saturated at N=50, and the test coverage was 1 at N=69, when the whole link failure test was completed. In all the NWs and Cases, the test coverage exceeded 50% when the test case count $N \leq 10$. Furthermore, in Figure 6 (ii), the test coverage in Case3 at NW6 was increased slowly compared with other NWs in Case3 because the number of link bandwidth is larger than that of other NWs.

These experimental results indicate that our proposed method is able to obtain the sufficient coverage even when the number of test count N is small.

D. Link attribute and Test Case Count

In the telecom network consists of the different types of equipment, such as switches, the telecom company decides on own test coverage to satisfy the link failure test. Therefore, we evaluate the number of test target attributes and test case count.

To evaluate it, all $w_{equipment\_type}$ are calculated using the equipment types, which are randomly assigned from 1 to 5 per k to each equipment at both sides of links in NW5. The result showed that the number of equipment types is from 12 to 79.

Figure 7 shows the number of test case count N and the number of equipment types $w_{equipment\_type}$ when the test coverage was varied from 25% to 100% at 25% intervals at NW5. From this graph, the test case count N is (almost) linearly increased as the number of equipment types. The increase ratio of the test case count N when the test coverage was 100%, was larger than that of 50%. When the number of equipment types was 79, the test coverage 100% was 3.1 times larger comparing of the test coverage 50%. For example, when the number of equipment types is 73 and the test coverage sets 75%, the telecom company needs to conduct the link failure test 35 cases.

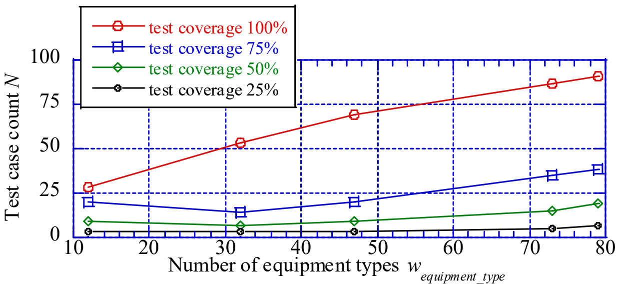


Figure 7. Number of equipment types $w_{equipment\_type}$ vs test case count N (ex. NW5).

From the above results, it indicates that if the target coverage sets higher, the number of test cases increases significantly.

V. CONCLUSION

In this paper, we proposed how to reduce link failure test cases by focusing on the topology similarity based on the idea of test case similarity using abstract path. We supposed a telecom network, where clients in the network, communicate with each other, and the link failure test assume single link failure. In this proposal, we clustered the possible test targets throughout a telecom network into groups of similar test targets based on the abstract path. We utilized topological similarity for the purpose, formulated a problem of minimizing the number of test cases, and conducted simulation experiments for 6 cases including large-scale networks with realistic topology. Through the experiments, we confirmed that our proposed method can drastically reduce the test case count. In future, we plan to verify the similarity based on the degree of the switch, propagation delay, and multiple failures.

REFERENCES

[1] M. Shibuya, H. Kawakami, T. Hasegawa, and H. Yamaguchi, “Design and implementation of support system for network testing with whitebox switches,” COLLA 2016, pp.39-44, Nov. 2016.

[2] http://www.sdnjapan.org/2015/1411_torii.pdf (in Japanese), [retrieved: May, 2018].

[3] http://www.okinawaopenlabs.org/wp/wp-content/uploads/20160205_tanabe.pdf (in Japanese), [retrieved: May, 2018].

[4] Mininet, http://mininet.org/, [retrieved: May, 2018].

[5] QualiSystems CloudShell / TestShell, http://www.qualisystems.com/products/cloudshell-add-ons/testshell-overview/, [retrieved: May, 2018].

[6] D. G. Corneil and C. C. Gotlieb, “An Efficient Algorithm for Graph Isomorphism,” Journal of the ACM, Vol. 17, Issue 1, pp.51-64, Jan. 1970.

[7] J. R. Ullumn, “An Algorithm for Subgraph Isomorphism,” Journal of the ACM, Vol. 23, Issue 1, pp.31-42, Jan. 1976.

[8] B. D. McKay, ”Practical Graph Isomorphism,” Congressus Numerantium, vol. 30, pp. 45-87, 1981.

[9] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento, ”A (Sub) Graph Isomorphism Algorithm for Matching Large Graphs,” IEEE Trans. Pattern Analysis and Machine Intelligenc, , vol. 26, no. 10, pp. 1367-1372, Oct. 2004.

[10] G. Kollias, S. Mohammadi, and A. Grama, “Network similarity decomposition (NSD): A fast and scalable approach to network alignment,” IEEE Transactions on knowledge and Data Engineering, Vol. 24, No. 12, pp.2232-2243, Dec. 2012.

[11] IBM ILOG CPLEX Optimiazation Studio, https://www.ibm.com/developerworks/downloads/ws/ilogcplex/, [retrieved: May, 2018].

[12] IIJ, https://www.iij.ad.jp/en/company/network/backbone/images/bawckbone_en.gif backbone_en.gif, [retrieved: May, 2018].

[13] WIDE, http://www.wide.ad.jp/About/report/pdf2016/usb/wide-memo-two-report-2016-01.pdf, [retrieved: May, 2018].