A Novel Method for Survivability Test Based on End Nodes in Large Scale Network

Ming Liang*, Zhao Gang, Wang Dongxia, Huang Minhuan, Li Xiang, Miao Qing and Xu Fei
National Key Laboratory of Science and Technology on Information System Security
Beijing, China
[e-mail: mingliang79@126.com; zg@public.bise.ac.cn; dongxiawang@126.com; hminwell@gmail.com; lixiang8358@hotmail.com; miaofj@163.com; xufei1023@126.com]
*Corresponding author: Ming Liang

Received January 23, 2014; revised October 25, 2014; accepted December 13, 2014; published February 28, 2015

Abstract

Survivability is a necessary property of network system in disturbed environment. Recovery ability is a key actor of survivability. This paper concludes network survivability into a novel composite metric, i.e. *Network Recovery Degree (NRD)*. In order to measure this metric in quantity, a concept of *Source-Destination Pair (SD Pair)*, is created to abstract end-to-end activity based on end nodes in network, and the quality of *SD Pair* is also used to describe network performance, such as connectivity, quality of service, link degree, and so on. After that, a Survivability Test method in large scale Network based on *SD pairs, called STNSD*, is provided. How to select *SD Pairs* effectively in large scale network is also provided. We set up simulation environment to validate the test method in a severe destroy scenario and evaluate the method scalability in different large scale network scenarios. Experiment and analysis shows that the metric *NRD* correctly reflects the effort of different survivability strategy, and the proposed test method *STNSD* has good scalability and can be used to test and evaluate quantitative survivability in large scale network.

Keywords: Network survivability, source-destination pair, network recovery degree, test method, survivability metric

http://dx.doi.org/10.3837/tiis.2015.02.008 **ISSN : 1976-7277**

1. Introduction

Survivability is a research hotspot in network area during recent twenty years, but how to measure network survivability accurately and quantitatively is still a challenge, especially in a large scale network. The generally accepted definition of network system survivability was proposed by Ellison et al. [1]. Survivability is the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents. The goal of network survivability is to maintain the fundamental network system services in the face of faults/failures and to support the fulfillment of organization missions. As for this definition, survivability is not mathematically well defined since this concept does not directly refer to a measurable sense. In order to solve this problem, this paper research to provide a novel metric and a test method for measuring network survivability in quantity, which help network survivability evaluation more accurately, experiment result shows the metric is valid and the method can work well in large scale network.

The rest of this paper is organized as follows: Section 2 reviews previous work on the network survivability test methods and metric. Section 3 analyzes the fundamental attributes of network survivability and the network survivability lifecycle, which is a general survivable process when a network suffering disturbances. Section 4 provides a novel quantifiable metric of network survivability, i.e. *Network Recovery Degree (NRD)*. Section 5 presents a method called *STNSD* to test the metric *NRD*, and section 6 makes an experiment and analysis to validate the proposed method *STNSD*. Finally section 7 presents conclusions and future work.

2. Related Work

Existing survivability test methods including survivability indexes for network system can be classified into two categories by different research ideas: effect-based survivability test methods [5][7][8] and scheme-based survivability test methods [9][10][11].

The idea of effect-based survivability test methods is analysing survivability definition of network system, abstracting key effect and characteristic of survivability for some requirements, and then providing survivability indexes, metrics and test method. These researchers think that survivability could be measured and evaluated using its quantitative attributes, especially the availability and fault-tolerance attributes which can be statistically modeled using the metrics of failure rate [2], repair rate [3], fault-coverage [4][5], and so on. Therefore survivability test is usually replaced by measuring its fundamental attributes which comes from survivability definition and effect. Under this methodology, Carmichael [5] analysed two kinds of typical Layer-2 Ethernet protocols available for increasing the survivability of a network on live networks and via simulation, and gave some survivability index in Layer-2, including forward delay, packet latency, Hello Time, Maxi age of packet, and so on. These works are made effectively on a small network with serval links in the experiment, but they are hard to work well in large scale network. Zuo [7] regarded survivability characteristic as adaptability, recoverability, fault tolerance, reliability, and performance degrading. And he has also listed 15 kinds of representative survivability primitives for these five characteristics, but he did not present quantifiable test method for these characteristics. Wang etc. [8] defined three quantifiable metrics: network robustness, cumulative failure probability and performance benefit to evaluate capacity allocation strategy for network survivability in the condition of limited redundant resources, and also gave corresponding calculation methods, but these methods only fit for survivability measurement of the largest connected component of network after failures, and does not fit for testing the whole network survivability obviously.

The idea of scheme-based survivability test methods is developing and measuring survivability index by means of analysing survivability policies or schemes. Zhang [9] proposes a new network survivability paradigm, called heterogeneous networking, for improving a network's defence capabilities. So they develop a quantifiable survivability metric, called diversity distance, to capture the extent of required redundancy and reduce the likelihood of network elements for each level of functional capability. Wang [10] views that the key policy of survivability is the dynamic reconfiguration. So many metrics, such as available battery power, varying communication bandwidth, available memory or faults in software components, should be considered in order to preserve desired application level quality of service, so he proposes corresponding methods for these metrics. Sasitharan [11] focuses on researching self-governance, such as self-organization, self-management, to support network survivability, and uses average link gradients and average node load as network survivability index, and propose bio-inspired mechanisms to support network survivability that is integrated with policy based management. All these methods above focuse on measuring survivability characters for some survivability schemes, such as diversity distance for the method in [9], available battery power for the method in [10], average node load for the method in [11], which are all not universal test methods in a general sense for different survivability mechnisms.

In a word, contrast with the survivability definition [1], the survivability metrics and test methods above are limited in network scale, or only measure some phrase(s) or aspect(s) of network survivability. So how to measure network survivability quantitatively in the whole and make a test method independent of mechnisms is still a challenge.

3. Network Survivability Lifecycle

In this section, we will analize the attributes of network survivability in detail, and make a theory preparation for proposing novel network survivability metric NRD.

Survivability is an important dynamic system attribute, and network survivable process can pass through five phases, which are normal phase, resistance phase, destroyed phase, recovery phase, and adaptation and evolution phase [12]. We call this five-phase process as a network survivability lifecycle. In the network survivability lifecycle, normal phase is a healthy and undestroyed stage; resistance phase is a degradation stage of network performance after attacks, failures, or accidents; destroyed phase is a bottom stage in which network performance lies at the lowest level; recovery phase is a rising stage of network performance since survivability mechanism runs; adaptation and evolution phase is a self-adaptation stage of network with the implementation of adaptation policies. When a disturbance is coming, the survivable process of network is shown in Fig. 1, where horizontal coordinate is time t , and vertical coordinate is network performance $V(t)$, and V_0 is the normal performance, and V_r is the required performance threshold of network system, and V_e is emergency performance threshold of network system. From Fig. 1, network is in normal phase initially, and $V(t)$ is at a high level. When network is disturbed in resistance phase, $V(t)$ degrades in evidence. With the further degradation of $V(t)$, network goes into destroyed phase and $V(t)$ reaches the lowest value (i.e. bottom level). Then, $V(t)$ rises gradually and network goes into recovery phase since survivability mechanism works. When self-adaptation policies work, network comes into evolution phase, during which the performance of network will become much better.

In a survivability lifecycle, the recovery time of network performance from phase 2 to 4 is a key index of network survivability, so we can grasp each phase by measuring the wave of network performance, and then get quantifiable survivability of a network by calculating the recovery time or analogous index in this survivable process.

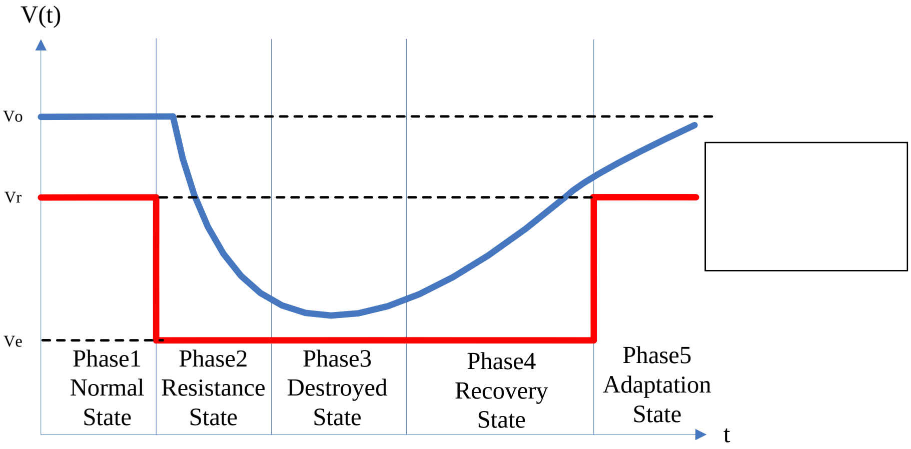


Fig. 1. Performance curve in a survivability lifecycle

4. Network Survivability Metric Based on SD Pair

Network survivability metric is the basic of the test and evaluation of network survivability. This section will set up two notions, *SD Pair* and *SD Quality*, to analyze the wave of network performance during a network survivability lifecycle, and then proposes a novel quantifiable metric of network survivability, i.e. *NRD*, based on *SD Pair*.

4.1 Analysis on Network Performance

According to the discussion in section 3, by means of analyzing the variety index of network performance, we can measure the survivability of network system in quantity. From our survey, methods that attempt to quantitatively analyze the survivability can be classified into two categories: connectivity or performance [13]-[15].

The analysis of network connectivity is based on two factors: the Node Connectivity Factor (NCF) and the Link Connectivity Factor (LCF). The former deals with the removal of nodes, while the latter is concerned with the removal of links. Several methodologies can be used to analyze the connectivity of networks, among which linear/non-linear programming and simulation with given metrics are the most popular [13].

In general, network performance is analyzed by calculating the capability that the network will fulfill its given QoS metrics. Because of the variety of network performance metrics, many diverse methodologies, such as Markov chain, game theory and simulation with given metrics, can be used for analysis [14].

We can easily find out that two kinds of analysis methods on network performance above are both concerned with the end-to-end activity in the network. In fact, the end-to-end activity is primary task of computer network and is also the most concern of users [15]. Users always rather care the ability of end-to-end activity than that in the midway. So it is an effective way to measure the performance of a large scale network system by means of analyzing the ability of end-to-end activity.

4.2 SD Pair and SD Quality

There are a lot of end-to-end activities in network system, such as a community between two nodes, a request between user and server or a web request between two web services, and so on. In order to describe these end-to-end activities and its capability, some concepts are defined as follows.

DEFINITION 1. *Source-Destination Pair*, for short *SD Pair*, is an abstraction of end-to-end activity of network system, denoted as $r = \{s, d\}$, $r \in R, s, d \in V$, where s and d denote the requester side and the target side in the end-to-end activity respectively, and R is the set of *SD Pairs*, and V is the set of nodes in the network.

According to the discussion in section 4.1, we categorized the *SD Pairs* to into two subcategories: the *Connectivity-type SD Pair* and the *Service-type SD Pair*. The *Connectivity-type SD Pair* is those *SD Pair* whose function focuses on network connectivity, and the *Service-type SD Pair* is those *SD Pair* whose function focuses on network application.

DEFINITION 2. *SD Quality*, denoted as p_r, is the degree to which end-to-end performance of *SD Pair* r fulfils connectivity or service requirement. *SD Quality* shows the capability of end-to-end activity of *SD Pair*.

By means of the two concepts *SD Pair* and *SD Quality*, we can make formal description for some general index of network performance, such as link degree, bandwidth, quality of service (QoS), and so on. In the *Connectivity-type SD Pair*, *S* and *D* denote two ends of the communication path separately, and *SD Quality* denotes the communication quality, such as bandwidth, delay, and so on. While in the *Service-type SD Pair, S* and *D* denotes the service requester and responder separately, and *SD Quality* denotes the service providing quality, such as service response time. In order to accurately describe the communication quality and service response quality, we further make another two definitions about *SD Quality*: the *Connectivity-type SD Pair* and the *Service-type SD Pair*, and give corresponding calculation method for each.

DEFINITION 3. *Connectivity-type SD Quality*, p_r, is the degree to which end-to-end performance of r fulfils connectivity requirement, and can be calculated by Formula (1):

$$p_r = \min(\frac{RTT_{reqr}}{RTT_{real}}, 1) \tag{1}$$

In Formula (1), RTT_{real} denotes the measured RTT (Round-Trip Time) in the Connectivity-type *SD Pair*, and RTT_{reqr} denotes the required RTT in the *Connectivity-type SD Pair*.

DEFINITION 4. *Service-type SD Quality*, p_r, is the degree to which end-to-end performance of r fulfils service requirement, and can be calculated by Formula (2):

$$p_r = \min(\frac{RST_{reqr}}{RST_{real}}, 1) \tag{2}$$

In Formula (2), RST_{real} denotes measured service response time in the *Service-type SD Pair*, and RST_{reqr} denotes the required service response time in the *Service-type SD Pair*.

In general, considering survivability measurement only for network communication system, *Connectivity-type SD Quality* is enough; while considering survivability measurement for network application system, *Service-type SD Quality* is much more suitable. With the definitions above, we can get a formal description of the network performance in the whole. Given a network, any end-to-end function can be denoted as r, and its performance can be denoted as p_r, then all the end-to-end function in the network can be denoted as $R = \{r_1, r_2, \ldots, r_N\}$, where N is the total number of the end-to-end functions, and all the end-to-end performance in the network can be denoted as $P_R = \{p_1, p_2, \ldots, p_N\}$. If we only consider the end-to-end performance in a large scale network, we can accurately analyze network performance and then measure network survivability by calculating P_R.

4.3 Network Survivability Metric Based on SD Pair

SD Quality denotes the end-to-end performance in the network, so, as discussion in section 3, *SD Quality* will wave in a survivability lifecycle. Now we will make another two definitions to further describe this survivable process, and then propose a composite metric, *NRD*, to measure network survivability in quantity.

DEFINITION 5. *SD Recovery Rate*, η_r, is the rate at which *SD Quality* is recovered. It can be calculated by Formula (3):

$$\eta_r = \frac{\mathrm{p_{recovery}}}{\mathrm{p_{normal}}} \tag{3}$$

In Formula (3), $\mathrm{p_{recovery}}$ denotes the *SD Quality* in recovery phase of survivable network, and $\mathrm{p_{normal}}$ denotes the *SD Quality* in normal phase of survivable network.

DEFINITION 6. *SD Recovery Time*, t_r, is the time from being destroyed to that 80% *SD Recovery Rate* is reached, which is calculated by Formula (4):

$$t_r = t^{\uparrow}{}_{80\%} - t^{\downarrow}{}_{80\%} \tag{4}$$

In Formula (4), 80% is a reference value that can be adjusted as necessary.

DEFINITION 7. *Network Recovery Degree (NRD)*. Given a network, each *SD Pair* r, $r \in R$, has a *SD Recovery Time* t_r, then the *NRD* of the network can be calculated by Formula (5):

$$NRD = 1 \Big/ \sum_{r \in R} \omega_r t_r \tag{5}$$

In Formula (5), ω_r is the weight of *SD Pair* r, which reflects the importance of r and can be got by experience, and t_r is the recovery time of *SD Pair* r, and R is the set of *SD Pairs*. In fact, *NRD* is a composite metric, which reflects how much performance of the whole network can be recovered from destroyed state in a unit of time, so the more *NRD* is, the better network survivability is.

5. Survivability Test Method in the Large Scale Network Based on SD Pairs

5.1 Test Idea

The Survivability Test method in the large scale Network based on *SD pairs*, called *STNSD*, firstly abstracts the network function with *SD Pair*, and describes connectivity and service performance with *SD Quality*. Then considering different failure scenes in a large scale network, we will make different test cases and use an active measurement tool to keep monitoring the RTT (Round-Trip Time) of each *SD Pair*, and then calculate *SD Recovery Rate* and *SD Recovery Time*. Finally, by using Formula (5), we can calculate the *NRD* and get quantifiable survivability of the network.

5.2 Test Steps

Under the test idea, we have detailed test steps of the method *STNSD* below.

Step 1: Construct failures. For *connectivity-type SD Pair*, the failure situation focuses on communication link or node whose effect is disconnection or link congestion. For *Service-type SD Pair*, the failure situation focuses on quality of service whose effect is Deny of Service or Congestion of Service.

Step 2: Select *SD Pairs* for survivability test, in which *SD Pairs* should pass though the affected area including compromised nodes, links and services. Sometimes key nodes, links, and services in the network also should be selected into *SD Pair* set. The selection of SD Pair is described in section 5.3 in detailed.

Step 3: Measure *SD Quality* of each *SD Pair* by Formula (1) or Formula (2).

Step 4: Measure *SD Recovery Rate* and *SD Recovery Time* for each *SD Pair* by Formula (3) and Formula (4), and then calculate the *NRD* of the whole network by using Formula (5), and network survivability is got in quantity.

5.3 Selection of SD Pair Set

The *SD Pairs* set can be selected according to the type of failure: for service failure, The *SD Pair* between the compromised server and a user node should be considered; for link failure, node failure, protocol error, etc., the *SD Pairs* should be selected by an effective and general approach for different failure positions and types in the network. The idea of selection of *SD Pair* set is selecting a part of *SD Pairs* to get the response of all the *SD pairs* based on end node so as to watch the performance of network inside as full as possible. By this means, we need not consider positions and number of destroyed area in the network, because all the failures which can affect end-to-end performance based on end nodes can be found by such optimal *SD Pairs*. This work can be translated to resolve a set cover problem, which is a NP-complete problem of Karp [16]. In this section, we propose an *Optimal SD Selection (OSDS)* algorithm based on end nodes for selecting *SD Pair* set in large scale network.

In order to describe this algorithm, we should define a *Cover-Degree Function* $f(\mathrm{R})$. The *Cover-Degree Function* reflects the degree with which the set R covers the network links. For each *SD Pair* set R, there exists:

$$f(\mathrm{R}) = \left|\mathrm{U}_{\mathrm{r} \subset R}\, r\right| / |\mathrm{L}| \tag{6}$$

In which, the r is a *SD Pair* in R, and $r = \{s, d\}$ where s and d are both end nodes. As a *SD Pair* cross through an end-to-end path, to simplify it we also define $r = \{\mathrm{s}, \mathrm{d}\} = \{(\mathrm{s}, \mathrm{x}), \ldots, (\mathrm{y}, \mathrm{d})\}$, where x,…,y are the nodes crossed through by *SD Pair* r, and $(\mathrm{s}, \mathrm{x}), \ldots, (\mathrm{y}, \mathrm{d})$ are a group of links in *SD Pair* path. L is network link set, and $|\mathrm{L}|$ is the total number of network links.

Given a set of end nodes $\mathrm{V}_{\mathrm{end}}$ in a network, the *OSDS* algorithm has four steps.

Step1: initialize *SD Pair* set R with empty, and set precision parameter *M* and balance parameter *N*. Generally let $M = |V_{\mathrm{end}}| / 5$ and N=10 by experience.

Step2: for each node s in $\mathrm{V}_{\mathrm{end}}$, select other N nodes, $d_1, d_2, \ldots, d_i, \ldots, d_{\mathrm{N}} \in V'$ and $\mathrm{s} \neq \mathrm{d_i}$ from $\mathrm{V}_{\mathrm{end}}$, and make up of *N* candidate *SD Pair* of node s: $r_1 = \{s, d_1\}$, $r_2 = \{s, d_2\}$,…,

$r_N = \{s, d_N\}$. Add each candidate *SD Pair*, r_1, r_2, …, r_N, to *SD Pair* set R respectively, and calculate the new *Cover-Degree Function* $f(\mathrm{R})$.

Step3: for end node s, select the *SD Pair* $r_{\max}$ which makes $f(\mathrm{R})$ be the largest one in the candidate *SD Pairs* of node s, and then add $r_{\max}$ to the *SD Pair* set R. If all the *N* candidate *SD Pairs* of node s cannot enlarge $f(\mathrm{R})$, then we regard node s as an invalid node, and compute the number of continuous invalid nodes.

Step4: When the number of continuous invalid nodes is below *M*, repeat Step2 and Step3, otherwise finish the algorithm. When algorithm is finished, R is the optimal SD Pair set that we required.

The *OSDS* algorithm is shown in **Algorithm 1**.

Algorithm 1 *OSDS()*

```
Input: $\mathrm{V}_{\mathrm{end}}$
Output: $R$
  $R \leftarrow \phi$;
  $V' \leftarrow \mathrm{V}_{\mathrm{end}}$;
  $M = |V_{\mathrm{end}}| / 5$;
  $N = 10$;
  $\mathrm{invalid\_node} = 0$;
  while $\mathrm{invalid\_node} < \mathrm{M}$ do
    for node $s$ in $V'$ do
      randomly select $d_1, d_2, \ldots, d_i, \ldots, d_N \in V'$, and $\mathrm{s} \neq \mathrm{d_i}$, set $r_i = (\mathrm{s}, \mathrm{d_i})$;
      Choose $r_{\max} = (\mathrm{s}, \mathrm{d_{max}})$ to maximize $f(R \cup r_i)$;
      set $R \leftarrow R \cup r_{\max}$;
    end for
    Calculate $\mathrm{invalid\_node}$ for the number of successive nodes which cannot increase $f(R)$;
  end while
  return $R$;
```

In *OSDS* algorithm above, the precision parameter *M* is used to adjust the precision of the algorithm. The larger *M* results in larger cover degree of *SD Pair* set. But meanwhile, the computation complexity of *OSDS* algorithm is enlarged evidently. The balance parameter *N* is used to adjust the size of *SD Pair* set and computing speed of algorithm. The more *N*, *OSDS* algorithm has larger probability to find new valid *SD Pairs*. On the other side, computation complexity is also enlarged evidently. But the computation complexity of *OSDS* algorithm is no more than O(M*N*$|V_{\mathrm{end}}|$). Experience shows that we can obtain acceptable calculation precision and speed, when *N=10* and $M = |V_{\mathrm{end}}| / 5$, where $|V_{\mathrm{end}}|$ is the total number of end nodes of network.

The *SD Pairs* above fall into different types: the *SD Pair* on server failure is *Service-type SD Pair*, and the *SD Pair* selected by *OSDS* algorithm are all *Connectivity-type SD Pair*. So these *SD Pairs* should use different formulas to calculate *SD Quality* in the test step 3 in section 5.2.

5.4 Weight of SD Pair

For the *Service-type SD*, the weight is assigned by experience, and the more important the path of *SD Pair* is and the larger weight is. For *Connectivity-type SD*, its weight is decided by its largest link bandwidth in the *SD Pair* path, the higher the link bandwidth of *SD Pair* is and the larger weight is. Formula (7) is given for calculate the weight, ω_i , of No. i *SD Pair*.

$$\omega_i = \frac{\log W_i}{\sum_{i=1}^{n} \log W_i} \tag{7}$$

In Formula (7), W_i is the largest link bandwidth (Mb/s) in No. i *SD Pair*, and n is the total number of *SD Pairs*.

6. Experiment and Analysis

The test experiment aims: (i) to assess the *STNSD* method's ability including *OSDS* algorithm for selecting *SD Pairs*; (ii) to validate the *STNSD* method to calculate quantifiable survivability in large scale network; (iii) compare the performance gain of the *STNSD* method facing different survivability strategy. For this purpose, a use case of the STNSD is given, and the scalability analysis of the STNSD method is also given.

6.1 Experiment Scenarios

To demo the test process of the STNSD method, we set up a network with 66 routers (Scenario 1) labeled by from n0 to n65 and simulated by CORE v4.3 [17] tools. Another five scenarios are also set up to analyze the scalability of the STNSD method in large scale network. Although the Scenario 1 is not large enough, it embraces the network core, network aggregation layer, and network access layer, and can be seen as a branch of large scale network. The topology of Scenario 1 includes four areas: one backbone area and three not-so-stubby areas shown in **Fig. 2.** Supposed 15 links (33% percent links in aggregation layer) of the network are destroyed, which lie in the three not-so-stubby areas randomly. The destroyed links are labeled by dotted lines in **Fig. 2.** In order to test the network survivability, we apply two kinds of strategies with different route protocols: *distributed route strategy* (Strategy 1) and *centralized route strategy* (Strategy 2). In *distributed route strategy* the new route table is calculated in each router by OSPF protocol after a failure, while in *centralized route strategy* the new route tables are calculated by some centralized servers and then spread them to routers, which is appeared in many research papers. For just giving the survivability test as a sample we do not describe the *centralized route strategy* in detail.

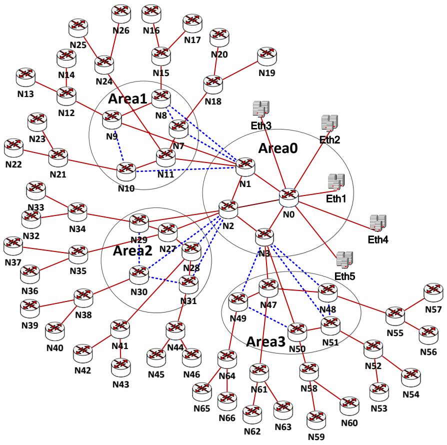
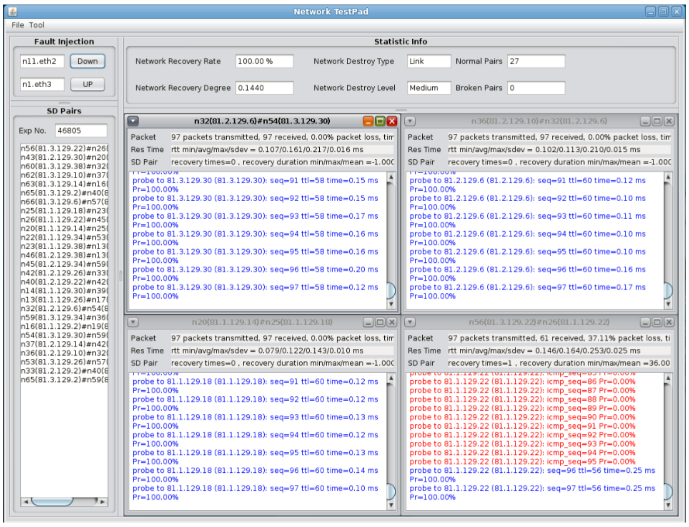


Fig. 2. Network topology in experiment Scenario 1

In order to assess scalability of the STNSD method, we also generate another 5 network scenarios with number of nodes from 145 to 1467 by a popular topology generator, GT-ITM. The nodes include both internal nodes and end nodes (leaf nodes).

- Scenario 2: network with 145 nodes (including 132 end nodes) and 147 links. Network run with OSPF routing protocol.
- Scenario 3: network with 322 nodes (including 277 end nodes) and 349 links. Network run with OSPF routing protocol.
- Scenario 4: network with 528 nodes (including 428 end nodes) and 635 links. Network run with OSPF routing protocol.
- Scenario 5: network with 712 nodes (including 559 end nodes) and 887 links. Network run with OSPF routing protocol.
- Scenario 6: network with 1467 nodes (including 1126 end nodes) and 1898 links. Network run with OSPF routing protocol.



Fault Injection

n11.eth2

n1.eth3

SD Pairs

Exp No. 46805

```
n56(81.3.129.22)#n26(
n43(81.2.129.30)#n20(
n60(81.3.129.38)#n32(
n62(81.3.129.10)#n37(
n63(81.3.129.14)#n16(
n65(81.3.129.2)#n40(8
n66(81.3.129.6)#n57(8
n25(81.1.129.18)#n23(
n26(81.1.129.22)#n45(
n20(81.1.129.14)#n25(
n22(81.1.129.34)#n53(
n23(81.1.129.38)#n13(
n46(81.2.129.38)#n13(
n45(81.2.129.34)#n59(
n42(81.2.129.26)#n33(
n40(81.2.129.22)#n42(
n14(81.1.129.30)#n39(
n13(81.1.129.26)#n17(
n32(81.2.129.6)#n54(8
n59(81.3.129.34)#n36(
n16(81.1.129.2)#n19(8
n54(81.3.129.30)#n59(
n37(81.2.129.14)#n42(
n36(81.2.129.10)#n32(
n53(81.3.129.26)#n57(
n33(81.2.129.2)#n40(8
n65(81.3.129.2)#n59(8
```

Statistic Info

Network Recovery Rate 100.00 %
Network Destroy Type Link
Normal Pairs 27
Network Recovery Degree 0.1440
Network Destroy Level Medium
Broken Pairs 0

n32(81.2.129.6)#n54(81.3.129.30)

```
Packet    97 packets transmitted, 97 received, 0.00% packet loss, tim
Res Time  rtt min/avg/max/sdev = 0.107/0.161/0.217/0.016 ms
SD Pair   recovery times=0 , recovery duration min/max/mean =-1.000
probe to 81.3.129.30 (81.3.129.30): seq=91 ttl=58 time=0.15 ms
Pr=100.00%
probe to 81.3.129.30 (81.3.129.30): seq=92 ttl=58 time=0.15 ms
Pr=100.00%
probe to 81.3.129.30 (81.3.129.30): seq=93 ttl=58 time=0.17 ms
Pr=100.00%
probe to 81.3.129.30 (81.3.129.30): seq=94 ttl=58 time=0.16 ms
Pr=100.00%
probe to 81.3.129.30 (81.3.129.30): seq=95 ttl=58 time=0.16 ms
Pr=100.00%
probe to 81.3.129.30 (81.3.129.30): seq=96 ttl=58 time=0.20 ms
Pr=100.00%
probe to 81.3.129.30 (81.3.129.30): seq=97 ttl=58 time=0.12 ms
Pr=100.00%
```

n36(81.2.129.10)#n32(81.2.129.6)

```
Packet    97 packets transmitted, 97 received, 0.00% packet loss, tim
Res Time  rtt min/avg/max/sdev = 0.102/0.113/0.210/0.015 ms
SD Pair   recovery times=0 , recovery duration min/max/mean =-1.000
probe to 81.2.129.6 (81.2.129.6): seq=91 ttl=60 time=0.12 ms
Pr=100.00%
probe to 81.2.129.6 (81.2.129.6): seq=92 ttl=60 time=0.10 ms
Pr=100.00%
probe to 81.2.129.6 (81.2.129.6): seq=93 ttl=60 time=0.11 ms
Pr=100.00%
probe to 81.2.129.6 (81.2.129.6): seq=94 ttl=60 time=0.10 ms
Pr=100.00%
probe to 81.2.129.6 (81.2.129.6): seq=95 ttl=60 time=0.10 ms
Pr=100.00%
probe to 81.2.129.6 (81.2.129.6): seq=96 ttl=60 time=0.16 ms
Pr=100.00%
probe to 81.2.129.6 (81.2.129.6): seq=97 ttl=60 time=0.17 ms
Pr=100.00%
```

n20(81.1.129.14)#n25(81.1.129.18)

```
Packet    97 packets transmitted, 97 received, 0.00% packet loss, tim
Res Time  rtt min/avg/max/sdev = 0.079/0.122/0.143/0.010 ms
SD Pair   recovery times=0 , recovery duration min/max/mean =-1.000
probe to 81.1.129.18 (81.1.129.18): seq=91 ttl=60 time=0.12 ms
Pr=100.00%
probe to 81.1.129.18 (81.1.129.18): seq=92 ttl=60 time=0.12 ms
Pr=100.00%
probe to 81.1.129.18 (81.1.129.18): seq=93 ttl=60 time=0.13 ms
Pr=100.00%
probe to 81.1.129.18 (81.1.129.18): seq=94 ttl=60 time=0.12 ms
Pr=100.00%
probe to 81.1.129.18 (81.1.129.18): seq=95 ttl=60 time=0.13 ms
Pr=100.00%
probe to 81.1.129.18 (81.1.129.18): seq=96 ttl=60 time=0.14 ms
Pr=100.00%
probe to 81.1.129.18 (81.1.129.18): seq=97 ttl=60 time=0.10 ms
Pr=100.00%
```

n56(81.3.129.22)#n26(81.1.129.22)

```
Packet    97 packets transmitted, 61 received, 37.11% packet loss, ti
Res Time  rtt min/avg/max/sdev = 0.146/0.164/0.253/0.025 ms
SD Pair   recovery times=1 , recovery duration min/max/mean =36.00
probe to 81.1.129.22 (81.1.129.22): icmp_seq=86 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=87 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=88 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=89 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=90 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=91 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=92 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=93 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=94 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): icmp_seq=95 Pr=0.00%
probe to 81.1.129.22 (81.1.129.22): seq=96 ttl=56 time=0.25 ms
Pr=100.00%
probe to 81.1.129.22 (81.1.129.22): seq=97 ttl=56 time=0.25 ms
Pr=100.00%
```

Fig. 3. The panel of network survivability test by the STNSD Method

6.2 A Use Case of STNSD Method

We realize a tool for the *STNSD* method, the panel is shown in **Fig. 3**. The tool can collect and monitor information for each *SD Pair*, and calculate the *NRD* in the whole network on different scenarios. We use the Scenario 1 as an example to demo the STNSD Method. For the Scenario 1, 22 pairs of nodes are selected using the *OSDS* algorithm: {n56,n25}, {n43,n60}, {n20,n46}, {n62,n22}, {n65,n40}, {n14,n32}, {n17,n54}, {n63,n37}, {n66,n57}, {n39,n42}, {n26,n23}, {n13,n16}, {n59,n53}, {n36,n33}, {n25,n19}, {n56,n45}, {n56,n54}, {n43,n37}, {n60,n65}, {n20,n17}, {n22,n14}, {n40,n32}. According to end-to-end bandwidth of each pairs, we calculate weights of all pairs by the Formula (6) in **Table 1**.

In the experiment, as all the pairs are *Connectivity-type SD Pairs except* {n56,n25}, we use an active measurement tool, like Ping command, to keep measuring the RTT (Round-Trip Time) of each *Connectivity-type SD Pair*. For *Service-type SD Pair* {n56,n25}, we use an request tool written with python language to open the url of web application server deployed in node 25, and measuring the service response time. In the test steps descripted in section 5.2, given $RTT_{reqr} = 50ms$ and $RST_{reqr} = 100ms$, *Connectivity-type SD Quality* is calculated by the Formula (1) and *Service-type SD Quality* is calculated by the Formula (2), and then the *SD Recovery Rate* and the *SD Recovery Time* of each *SD Pair* are achieved by the Formula (3) and the Formula (4) respectively, As all the pairs, except {n66,n57}, {n26,n23}, {n13,n16}, {n59,n53}, {n36,n33}, {n25,n19} and {n43,n37}, in both the *distributed route strategy* and the *centralized route strategy* are recovered completely, so $\eta_r = 100\%$ for each pair. We repeat the test 10 times and calculate average value of each pair. The result of two survivability strategy is shown in **Table 2**. Because the *SD Pair* {n66,n57}, {n26,n23}, {n13,n16}, {n59,n53}, {n36,n33}, {n25,n19} and {n43,n37} are not affected by 15 destroyed links, so the two survivability strategies have no effect on network, and the value of the *SD Recovery Rate* and *SD Recovery Time* are all zero in the **Table 2.**

Table 1. *SD Pair*s and its Weights

No.	SD Pairs	Largest bandwidth	Weight
1	{n56,n25}	10G	0.0541
2	{n43,n60}	10G	0.0541
3	{n20,n46}	10G	0.0541
4	{n62,n22}	10G	0.0541
5	{n65,n40}	10G	0.0541
6	{n14,n32}	10G	0.0541
7	{n17,n54}	10G	0.0541
8	{n63,n37}	10G	0.0541
9	{n66,n57}	1G	0.0405
10	{n39,n42}	1G	0.0405
11	{n26,n23}	1G	0.0405
12	{n13,n16}	1G	0.0405
13	{n59,n53}	1G	0.0405
14	{n36,n33}	1G	0.0405
15	{n25,n19}	1G	0.0405
16	{n56,n45}	1G	0.0405
17	{n56,n54}	1G	0.0405
18	{n43,n37}	1G	0.0405
19	{n60,n65}	1G	0.0405
20	{n20,n17}	1G	0.0405
21	{n22,n14}	1G	0.0405
22	{n40,n32}	1G	0.0405

Table 2. *SD Pair*s and its η_r and t_r

No.	SD Pairs	Strategy 1		Strategy 2	
		t_r (s)	η_r	t_r (s)	η_r
1	{n56,n25}	33.779	100%	9.465	100%
2	{n43,n60}	41.375	100%	7.029	100%
3	{n20,n46}	51.387	100%	12.346	100%
4	{n62,n22}	69.407	100%	12.446	100%
5	{n65,n40}	39.374	100%	7.677	100%
6	{n14,n32}	41.042	100%	11.345	100%
7	{n17,n54}	49.718	100%	11.013	100%
8	{n63,n37}	40.040	100%	8.008	100%
9	{n66,n57}	0	100%	0	100%
10	{n39,n42}	38.716	100%	9.019	100%
11	{n26,n23}	0	100%	0	100%
12	{n13,n16}	0	100%	0	100%
13	{n59,n53}	0	100%	0	100%
14	{n36,n33}	0	100%	0	100%
15	{n25,n19}	0	100%	0	100%
16	{n56,n45}	40.376	100%	11.014	100%
17	{n56,n54}	40.379	100%	11.012	100%
18	{n43,n37}	0	100%	0	100%
19	{n60,n65}	38.039	100%	9.010	100%
20	{n20,n17}	38.374	100%	9.013	100%
21	{n22,n14}	59.061	100%	12.683	100%
22	{n40,n32}	41.042	100%	7.009	100%

In the test, the network survivability metric *NRD* is got by the Formula (5) using the data above, and the *NRD* of the *distributed route strategy* is 0.0315 while the *NRD* of the *centralized route strategy* is 0.1413. For a large scale network, generally *NRD* in [0.1, +∞) means excellent survivability, and *NRD* in [0.033, 0.1) means good survivability, and *NRD* in [0.02, 0.033) means acceptable survivability, and *NRD* in (-∞, 0.033) means bad survivability. So in this scenario the network with the *distributed route strategy* has an acceptable survivability, while the network with the *centralized route strategy* has an excellent survivability. It is obviously that much more performance of network is recovered by the *centralized route strategy*, so the *centralized route strategy* brings network better survivability than the *distributed route strategy* does in this scenario.

After analyzing the difference between two recovery strategies, we find out because route information update much faster in the *centralized route strategy* than that in the *distributed route strategy*, which makes much more *SD Pairs* come back to communication soon, so the *NRD* of the *centralized route strategy* is larger than that of the *distributed route strategy*. From experiment result, we can see that the *STNSD* method can be used to test network survivability well and the proposed metric *NRD* can perfectly reflect network survivability in a quantitative way.

6.3 Scalability Analysis

(1) Experiment Methodology

In large scale network, scalability of the method *STNSD* is very important. After analyzing the steps of the method *STNSD*, the *OSDS* algorithm is the key of the method *STNSD* to achieve good scalability for selection of *SD Pairs*. So we use the Scenario 1, 2, 3, 4, 5 and 6 which have different node number from 64 to 1467 to assess the scalability of *OSDS* algorithm.

For comparison, except for *OSDS* algorithm, another three algorithms: *all SD Pair* algorithm, *random SD Pair* algorithm, and prediction algorithm [18], are all considered. In *all SD Pair* algorithm, all end-to-end *SD Pairs* in the network are selected into *SD Pair* set. In *random SD Pair* algorithm, a number of *SD Pairs* are selected randomly into *SD Pair* set. In prediction algorithm [18], about 20% of *SD Pairs* in the network are considered based on algebraic prediction.

(2) Metrics

In order to evaluating the scalability of *OSDS* algorithm including its other performance, three metrics are used: cover rate, time cost, and SD Pair rate.

Cover rate reflects the validity of the *SD Pair* selection algorithm. Let the whole end-to-end SD Pair set in a network is R_{all}, the *SD Pair* subset obtained by the *SD Pair* selection algorithm is R , then the cover rate of the *SD Pair* selection algorithm is calculated by Formula (8):

$$cover\_rate = f(R) / f(R_{all}) \quad (8)$$

where $f(R_{all})$ and $f(R\ )$ are *Cover-Degree Function* of *SD Pair* set R_{all} and R respectively, which is already defined in section 5.3.

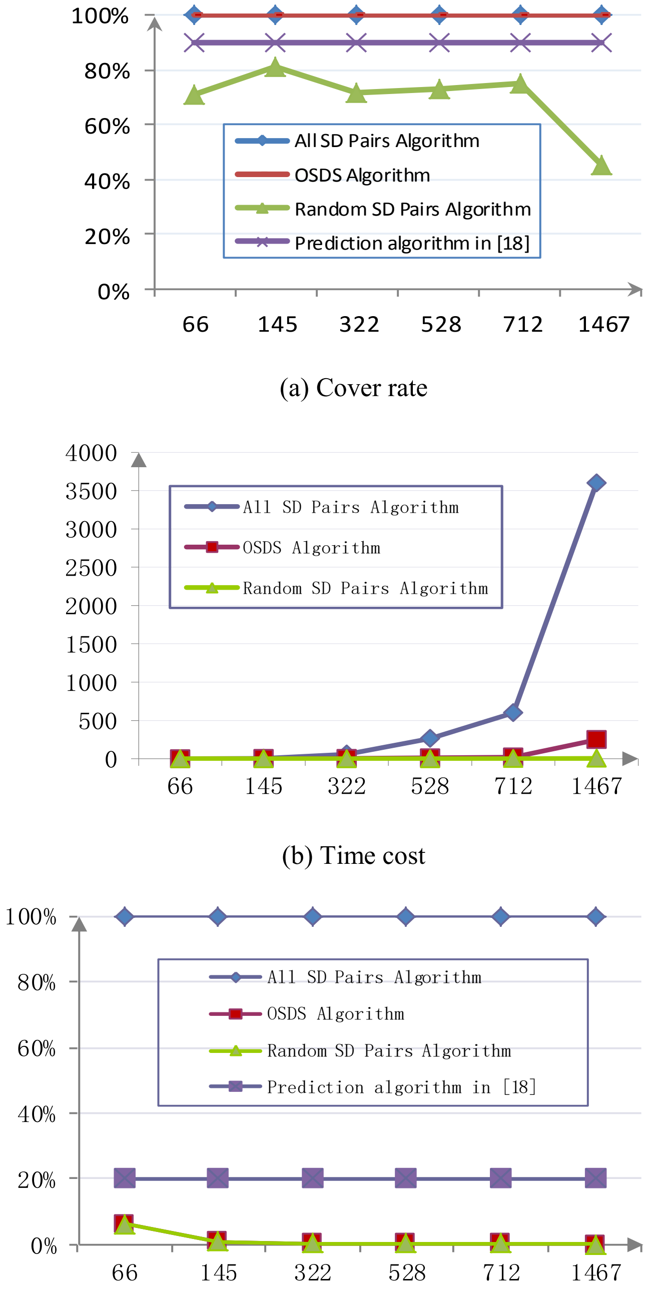
Time cost is cost time of the *SD Pair* selection algorithm, which reflects the efficiency of the algorithm.

SD Pair rate is the number of selected *SD Pairs* compared to the number of the total *SD Pairs* in the network. Let N be the number of the total *SD Pairs*, N_p be the number of SD Pairs that are obtained by the *SD Pair* selection algorithm. SD Pair rate is calculated by Formula (9).

$$SD\_Pair\_rate = N_p / N \tag{9}$$

(3) Experiment Results

The metric results of *OSDS* algorithm, *all SD Pair* algorithm, *random SD Pair* algorithm and prediction algorithm [18] in five scenarios are shown in **Fig. 4**.



(a) Cover rate

(b) Time cost

(c) SD Pair rate

Fig. 4. The result of metrics by four algorithms in different scale networks (x axis is node number of network, and y axis is cover rate, time cost, and SD Pair rate separately)

In **Fig. 4**, the horizontal axis is the node number of network according to the five scenarios, and the vertical axis is three metrics: cover rate, time cost, and SD Pair rate separately. **Fig. 4 (a)** shows the cover rate by four algorithms in the different topologies. From **Fig. 4 (a)**, we can clearly find out that cover rate of *OSDS* algorithm and *all SD Pair* algorithm always have the top value with 100%, while the other two algorithms are evidently lower. **Fig. 4 (b)** shows the time cost by three algorithms in the different topologies. From **Fig. 4 (b)**, we can clearly find out that the time cost by *OSDS* algorithm and *random SD Pair* algorithm both have much less time than *all SD Pair* algorithm. Even in Scenario 6 with 1467 nodes, the time cost is 248.60s, which is also acceptable. For lack of enough information, the time cost of prediction algorithm is unknown for us [18]. **Fig. 4 (c)** shows the SD Pair rate by four algorithms in the different topologies. From **Fig. 4 (c)**, we can clearly find out that SD Pair rate by *OSDS* algorithm is much lower than *all SD Pair* algorithm and prediction algorithm [18]. Even in Scenario 6 with 1467 nodes, the SD Pair rate of *OSDS* algorithm is as low as 0.12% (752 SD Pairs), which is much less contrast to other algorithms. For comparison, we also select the same number of SD Pairs by *random SD Pair* algorithm. We also find a decreasing trend of SD Pair rate with the increase of network size in **Fig. 4 (c)**. After analyzing the selected *SD Pairs* by *OSDS* algorithm, we also find out number of *SD Pairs* is no more than number of end nodes and there are less than 2 *SD Pairs* correspond to an end node, which pay little overload on network traffic and end nodes. What's more, because end-to-end path is mainly necessary for *OSDS* algorithm, *OSDS* algorithm can work well without the information of network topology, when using a tool like Traceroute. When all end-to-end paths are known, the computing complex is O(n) where n is the number of network nodes, so *OSDS* algorithm is scalable and fits large scale network.

A problem of SD Pair selection algorithm, including *OSDS* algorithm, based on end nodes is that when there are lots of redundant links, such as in Scenario 3, 4, and 5, the SD Pair paths based on end nodes might not cover all the links of network, but this could not affect network survivability test eventually. It's because: a) network survivability test is user-oriented. The end-users just care about the performance they can obtain, and need not to concern with network internal information. The uncovered links are always redundant, and usually do not work. b) The SD Pair paths have adaptability in the network. When link fault occurs, redundant links would transmit data, and would be immediately covered.

Experiment result and analysis shows that the proposed survivability test method *STNSD* using *OSDS* algorithm has an excellent scalability and can be used in large scale network.

6.4 Comparison Analysis between STNSD and Existing Survivability Test Methods

As described related works in section 2, existing survivability test methods have two categories by different research ideas: effect-based survivability test methods and scheme-based survivability test methods. Since the scheme-based methods are all related with corresponding survivability schemes closely, they cannot be used in survivability measurement with other schemes always. While the *STNSD* is a general method, which can be used to measure different survivability schemes for large scale network freely. What's more, the *STNSD* also resolves the problems about scale and quantization limitation of existing effect-based survivability test methods, and present a quantifiable test approach for a novel metric *NRD* and can be used in large scale network. In a word, the *STNSD* has a better performance compared to existing survivability test methods.

7. Conclusion and Future Work

This paper proposes a novel metric *NRD* on network survivability and its test method *STNSD* based on *SD Pairs*. The metric *NRD* is based on the abstraction of end-to-end capability of network, which can effectively reflect the network performance in the whole. The survivability test method *STNSD* can be used in quantifiable survivability test and evaluation for the large scale network. Experiment result shows that the survivability test method is valid and *NRD* can reflect the network survivability ability correctly, which makes survivability test and evaluation more efficiently. Because *OSDS* algorithm for selecting *SD Pairs* has well scalability for network size and is independent of the failure position, so the proposed survivability test method *STNSD* can be used to measure survivability in large scale network even when the failure position is unknown.

The proposed technique is simple and easy to apply in network measurement points. Presently a test framework is under development aiming to facilitate the technique usage on operational scenarios in real environment.

Acknowledgment

This work has been partially supported by the National Natural Science Foundation of China under Grant No. 61271252.

References

[1] Ellison B, Fisher D A, Linger R C, et al., “Survivable Network Systems: An Emerging Discipline,” CMU/SEI-97-TR-013, Pittsburgh: Carnegie Mellon University, 1997. Article (CrossRef Link)

[2] Cao C., Zukerman M., Wu W., Manton J. H. and Moran B., “Survivable Topology Design of Submarine Networks,” *Journal of Lightwave Technology*, 31(5): pp.715-730, 2013. Article (CrossRef Link)

[3] Lin F.Y.-S., Yu-Shun Wang, Hui-Yu Chung and Jia-Ling Pan, “Maximization of Network Survivability under Malicious and Epidemic Attacks,” in *Proc. of 2012 26th International Conference on Advanced Information Networking and Applications Workshops*, pp.412-417, 2012. Article (CrossRef Link)

[4] Howard F.Lipson, “Survivability: A New Security Paradigm for Protecting Highly Distributed Mission Critical Systems,” in *Proc. of the 38th Meeting of IFIP Working Group on Dependable Computing and Fault Tolerance*, June 28-July 2, 2000. Article (CrossRef Link)

[5] M. Al-Kuwaiti, N. Kyriakopoulos and S. Hussein, “A Comparative Analysis of Network Dependability, Fault-tolerance, Reliability, Security, and Survivability,” *IEEE Communcations Surveys&Tutorials*, 11(2), pp.106-124, 2009. Article (CrossRef Link)

[6] Carmichael L.S., Ghani N. and Rajan P.K., “Characterization and comparison of modern layer-2 Ethernet survivability protocols,” in *Proc. of the Thirty-Seventh Southeastern Symposium on*, pp.124- 129, 2005. Article (CrossRef Link)

[7] Yanjun Zuo, “A Framework of Survivability Requirement Specification for Critical Information Systems,” in *Proc. of the 43rd Hawaii International Conference on System Sciences*, pp.1-10, 2010. Article (CrossRef Link)

[8] Lina Wang, Furong Zhou, Chi Guo, Xiaoying Zhang and Mo Yang, “A Capacity Optimization Algorithm for Network Survivability Enhancement,” *IEEE International Conference on Multimedia Information Networking and Security*, pp.177-181, 2009. Article (CrossRef Link)

[9] Yongguang Zhang, Harrick Vin and Lorenzo Alvisi, “Heterogeneous Networking: A New Survivability Paradigm,” in *Proc. of ACM NSPW*, pp.33-39, 2001. Article (CrossRef Link)

[10] Wang Li, Li Zhi-Shu and Yin Feng, “A Dynamic Survivability Reconfiguration Framework Based on QoS,” *IEEE International Conference on Advanced Computer Control*, pp.103-106, 2008.

Article (CrossRef Link)

[11] Sasitharan Balasubramaniam, Dmitri Botvich, William Donnelly and John Strassner, "A Biologically Inspired Policy Based Management System for Survivability in Autonomic Networks, Broadband Communications, Networks and Systems," in *Proc. of BROADNETS Fourth International Conference on*, pp.160-168, 2007. Article (CrossRef Link)

[12] Liang Ming and Dongxia Wang, "Research on Test Model of Network Survivability," *Journal of Computational Information Systems*, 6(4), pp.1301-1309, 2010. Article (CrossRef Link)

[13] Y.-S. Lin, P.-H. Tsang, C.-H. Chen, C.-L. Tseng and Y.-L. Lin, "Evaluation of Network Robustness for Given Defense Resource Allocation Strategies," in *Proc. of the First International Conference on Availability, Reliability and Security*, 2006. Article (CrossRef Link)

[14] J.C. Knight, E. A. Strunk and K J. Sullivan, "Towards a rigorous definition of information system survivability," in *Proc. of DARPA Information Survivability Conference and Exposition*, Washington, DC, pp.78-89, 2003. Article (CrossRef Link)

[15] Frank Yeong-Sung Lin, Hong-Hsu Yen and Pei-Yu Chen, "Maximization of Network Survivability Considering Degree of Disconnectivity," in *Proc. of ICCSA 2011*, Part I, LNCS, vol. 6782, pp.667-676. Springer, Heidelberg , 2011. Article (CrossRef Link)

[16] Richard M. Karp, "Reducibility Among Combinatorial Problems". In R. E. Miller and J. W. Thatcher (editors). Complexity of Computer Computations. New York: Plenum. pp. 85-103, 2013. Article (CrossRef Link)

[17] Common Open Research Emulator (CORE), 2013. Article (CrossRef Link)

[18] Shun-an Wu, Qiao Yan, Xue-song Qiu and Yanjie Ren, "A Probe Prediction Approach to Overlay Network Monitoring," in *Proc. of 2011 7th International Conference on Network and Service Management*, pp.1-5, Oct.24-28, 2011, Paris. Article (CrossRef Link)

Ming Liang received his B.Sc., M.Sc. and Ph.D. degrees in Computer Science Dep artment from Mechanical Engineering College, China, in 2001, 2004 and 2008, respe ctively. Now he is a research assistant in National Key Laboratory of Science and Te chnology on Information System Security, China. His major interests are network sur vivability, mobile wireless network, and social network.

Zhao Gang received his Ph.D. degree in Computer Science from Tsinghua Univers ity, China, in 2009. Now he is a research fellow in National Key Laboratory of Scien ce and Technology on Information System Security, China. His main research interests include computer network, and information security.

Wang Dongxia received her M.Sc. and the Ph.D. degrees in Computer Science fro m the National University of Defense Technology, China, in 1997 and 2000, respecti vely. Now she is a research fellow in National Key Laboratory of Science and Techn ology on Information System Security, China. Her major interests are network surviv ability, network science, and mobile wireless networks.

Huang Minhuan received his B.Sc. and M.Sc. degrees from National University of Defense Technology, China, in 1996 and 1999, respectively. Now he is a research fellow in National Key Laboratory of Science and Technology on Information System Security, China. His major interests are network science, and Internet routing.

Li Xiang received her B.Sc. and M.Sc. degrees from Information Engineering University, China, in 2005 and 2008, respectively. Now she is a research assistant in National Key Laboratory of Science and Technology on Information System Security, China. Her current research interests include delay tolerant network, and mobile wireless network.

Miao Qing received her B.Sc. and M.Sc. degrees from National University of Defense Technology, China, in 1994 and 1997, respectively. Now she is a research assistant in National Key Laboratory of Science and Technology on Information System Security, China. Her current research interests include network science, and mobile wireless network.

Xu Fei received his B.Sc. and M.Sc. degrees from National University of Defense Technology, China, in 2005 and 2008, respectively. Now he is a research assistant in National Key Laboratory of Science and Technology on Information System Security, China. His current research interests include network science, and mobile wireless network.