Multi-layer fault diagnosis method in the Network Virtualization Environment

Congxian Yan, Ying Wang, Xuesong Qiu, Wenjing Li, Lu Guan
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, P.R.China, 100876

yancongxian1423@163.com, {wangy, xsqiu, wjli, guanlu}@bupt.edu.cn

Abstract—The performance and reliability of services relies on the network virtualization environment's capabilities to effectively detect and diagnose faults in both substrate and virtual network. However, Network Virtualization Environment (NVE) brings to fault diagnosis new challenges such as inaccessible substrate network information and multi-layer faults. To solve the above issues, a Multi-layer Fault Diagnosis Method (MFDM) is proposed. A layer-by-layer strategy is used to resolve the problem of inaccessible substrate network information. And a filtering algorithm is proposed to distinguish the multi-layer faults in the network virtualization environment. At last, a contribution-based hypothesis selection algorithm is proposed to infer the most possible faults. Simulations and experimental results show that MFDM has a higher performance in the accuracy ratio, false-positive ratio.

Keywords—fault diagnosis; Bayesian network; network virtualization environment; uncertainty reasoning

I. INTRODUCTION

Network Virtualization Environment (NVE) is a collection of multiple heterogeneous virtual networks from different service providers, which share the substrate network by the abstraction and isolation of substrate network resources [1]. In the NVE, the Internet Service Providers (ISPs) are divided into two entities: Infrastructure Providers (InPs) and Service Providers (SPs). SPs provide customized end-to-end services for users. InPs lease the substrate network resources to the SPs for building virtual networks, which improves the resource utilization of substrate network. The network virtualization technology keeps the physical level details from SPs and provides technical possibility for the running of diversity protocols and applications in the network. It also avoided the cost of operation and repeated purchases of infrastructure through the sharing of substrate infrastructure. For the foreseeable future, more and more network protocols and applications will run on the virtual networks. Thus the network robustness and survivability is vital in the evolution of virtual network.

Fault diagnosis is an important field in the research of network management. The goal of fault diagnosis is to locate network faults timely and accurately, thus to reduce the effect of faults on the services. However, the features of NVE bring the following challenges for fault diagnosis.

1) Inaccessibility of substrate layer information: Since InPs and SPs are different entities, InPs are generally unwilling to share the detailed network information with SPs. Such information includes the mapping relationships between virtual network and substrate network, the status of underlying network, the prior probability of underlying network, and so on, which is critical for fault diagnosis.
2) Multi-layer faults: Owing to the mapping relationship between substrate network and virtual network, some Substrate Faults (SFs) will lead to the failure of corresponding virtual nodes and links, which are called Correlative Virtual Faults (CVFs) in this paper. As for the faults occur because of the software errors in the virtual network, we call them Independent Virtual Faults (IVFs). Therefore, there are three types of faults in the NVE: SFs, CVFs, and IVFs. When SFs in the substrate network are repaired, their related CVFs will be automatically repaired. So we could repair the SFs and CVFs in the fault recovery. In order to provide convenience for fault recovery, we need to distinguish the IVFs from CVFs.

There are some diagnostic approaches in the field of network virtualization environment. Pan [1] proposed a probabilistic inference method which is based on the dependency between SFs and the observed symptoms. Zhang [2] proposed a service fault diagnosis algorithm which is based on inherent correlation among symptoms. However, these methods infer the faults based on the dependencies between SFs and symptoms, the existence of virtual faults are ignored in the process of fault inference. Tang [3] proposed a diagnostic approach which is based on the D-S evidence theory [4]. Steinder [5] provides a multi-layer fault diagnosis method which is based on the bipartite dependency graphs. However, these methods cannot distinguish CVFs from IVFs.

In order to solve the above problem, we propose a fault diagnosis method called MFDM. The main contributions are as follows:

1) A layer-by-layer strategy is proposed to deal with the inaccessible substrate network information. Firstly, virtual faults are inferred based on the symptom-virtual fault causality and the observed symptoms. Then substrate faults are located based on the mapping relationship between virtual faults and substrate faults. Based on the layer-by-layer diagnostic mode, the method can overcome the impact of inaccessible substrate network information
2) A filtering method is proposed to distinguish CVFs from IVFs. This method can inversely infer CVFs based on the mapping relationship between CVFs and SFs.

The rest of this paper is organized as follows. Section II describes the fault diagnosis problem of NVE. In section III, a fault diagnosis method MFDM which can locates SFs, VFs and distinguish IVFs from CVFs is introduced. In section IV, the experimental environment and results are presented and analyzed. Section V discusses the related work. In section VI, we conclude this paper and discuss the future work.

This work was supported by Fundamental Research Funds for the Central Universities **(BUPT 2013RC1103)**

II. PROBLEM FORMULATION

The mostly existing fault diagnosis methods are based on the dependence between symptoms and faults. In order to describe the relationship between symptoms and faults, a Bayesian network is built in the NVE.

In this paper, Virtual Components (VCs) mean the virtual nodes and links which may break down. Substrate Components (SCs) represent the physical substrate nodes (e.g. Servers, Routers) which may fail. Virtual network symptoms represent the statuses of the observed end-to-end services, which are made up of some VCs and can reflect the performance of the corresponding VCs. There exist mapping relationships between VCs and SCs in the VNE. A virtual node could be embedded into a substrate node. And a virtual link could be embedded into a substrate path, which is composed of several substrate nodes and links. So the status of VC can reflect the status of the SCs which is closely related with this VC.

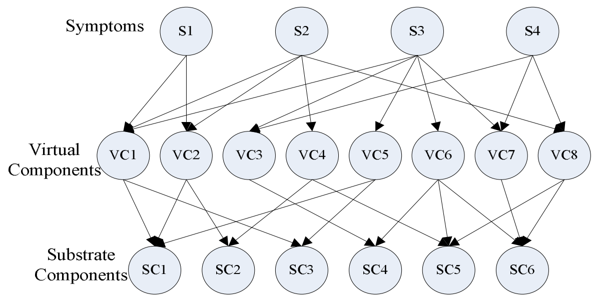


Fig.1.Bayesian network of a network virtualization environment

Figure 1 shows a three layered Bayesian network of a NVE instance. The first layer contains the observed virtual symptoms (e.g. S1, S2). The second layer consists of VCs that depend on SCs in the third layer. And the third layer is made up of SCs. The edges between the first layer and the second layer represent the dependency between VCs and Symptoms, which are weighted by the conditional probability distribution P ($VC_j|parent(VC_j)$). Parent (VC_j) means the parent nodes of VC_j (j=1, 2, 3…) (nodes pointing to VC_j in the Bayesian network) [6]. The edges between the second layer and the third layer represent the dependency between VCs and SCs, which can be weighted by P ($SC_i|parent(SC_i)$) (i=1, 2, 3…).

Then the fault diagnosis for the NVE turns into a probabilistic inference problem. After given a set of observed virtual symptoms, the most likely faults set (i.e. IVFs and SFs set) will be found, which can explain all the observed symptoms. In this paper, we propose a multi-layer fault diagnosis method based on the dependency between symptoms and faults.

The notations in Table.1 will be used in the process of fault diagnosis.

Table.1. Fault diagnosis notation

Notation	Definition
$Sum_{S_{of}}$	the number of all observed negative symptoms which can be explained by component f
Sum_{S_f}	the number of possible negative symptoms which can be explained by f
S	a set of all the symptoms
S_o	a set of all observed symptoms
F'	a set of all filtered components
α_0	the threshold in the filtering algorithm
F_s	a set of all the faults which can explain the symptom s
S_r	a set of all uncovered symptoms
S'	a set of all covered symptoms
S_{O_c}	a set of all observed symptoms which can be explained by the component c
S_c	a set of all the symptoms which can be explained by the component c
C_s	a set of all the components which can explain the symptom s

III. FAULT DIAGNOSIS ALGORITHM

In this section, we introduce a fault diagnosis method called Multi-layer Fault Diagnosis Framework (MFDM), which contains virtual fault diagnosis and substrate fault diagnosis. Firstly, it infers the VFs in the virtual networks based on the correlation between observed virtual symptoms and virtual faults. Secondly, it distinguishes IVFs from CVFs and regards CVFs as part of the observed substrate symptoms. Thirdly, it diagnoses the SFs in the substrate networks. In case the number of CVFs is not enough, some observed substrate symptoms are needed. In this paper, we adopt the same fault diagnosis approach in the virtual and substrate networks, which is called Improved Fault Diagnosis Approach (IFDA).

IFDA includes two sub-algorithms: filtering algorithm and maximum probable explanation algorithm. The filtering algorithm filters spurious faults and symptoms. The maximum probable explanation algorithm finds a most likely faults set to explain the observed symptoms. IFDA is based on the following assumptions:

1) Each fault will result in the happening of a symptom. In addition, each symptom can be explained by one fault.

2) All the faults are independent in the same network layer. It means that a fault cannot lead to the occurrence of another fault in the same network layer.

3) The probability of several faults occur simultaneously is low [7] [8].

4) If a negative symptom is received, there is at least one fault in the diagnosed path. If a positive symptom is received, all the components passing through the diagnosed path can be seen as the normal components [9][10].

A. Filtering algorithm

Some observed symptoms may be changed due to the noisy and dynamic environment. These spurious symptoms cannot accurately reflect the states of their associated components. This will make it difficult for fault inference. For instance, if the observed symptoms S1 and S3 indicate that a virtual node VC1 breaks down, while another symptom S2 reflects the virtual node VC1 is normal. We cannot distinguish the spurious symptoms from the associated symptoms of VC1.This will have an obvious influence on fault diagnosis. Since the probability of spurious symptoms is very low, the number of the spurious symptoms is small. In this paper, we propose a filtering algorithm to remove the influence of noise.

In addition, in this paper, the filtering algorithm is also used to distinguish CVFs from IVFs in the process of substrate fault diagnosis. All the VFs which are inferred in the virtual fault diagnosis approach will be regarded as part of the substrate symptoms in the substrate fault diagnosis. After filtering, the spurious symptoms are regarded as the IVFs, while the others are seen as the CVFs.

In order to filter the spurious symptoms, a ratio α is defined as follows:

$$\alpha = Sum_{S_{of}} \big/ Sum_{S_f},\ \alpha \in [0, 1] \qquad (1)$$

Where $Sum_{S_{o_f}}$ represents the number of all observed negative symptoms which can be explained by the component f. Sum_{S_f} means the number of possible negative symptoms which can be explained by f. All of these symptoms should be observed, but some of these symptoms may not be observed due to the dynamic network environment. The pseudo code of our algorithm is shown in Algorithm 1:

Algorithm 1: Filter spurious faults algorithm
Input: symptom set S, observed symptom set S_o, component candidate set F
Output: filtered component set F'
$F' = \{\}$;
for f ∈ F **do**
compute the observed symptoms number $Sum_{S_{o_f}}$ and the symptoms number Sum_{S_f};
α = $Sum_{S_{o_f}} / Sum_{S_f}$;
if α≥ α_0
add f into F';
end if
end for //select filtered components
for s∈ S_o **do**
find the components Fs which can explain the symptom s;
if Fs ∩ $F' = \emptyset$
remove s from S_o;
end if
end for //remove spurious symptoms
return F';

Algorithm 1: Filter spurious faults algorithm

In algorithm 1, the observed symptoms number $Sum_{S_{o_f}}$ and the symptoms number Sum_{S_f} for each component f are computed, and then its rate □ is calculated. If □< α_0, the component f will be regarded as a spurious fault (i.e. f=0) and abandoned. However, if □≥ α_0, f will be seen as a suspicious fault component and putted into F. For instance, in Fig.1 virtual component VC_1 breaks down. It will leads to the failure of the service S1, S2 and S3. If we receive a negative symptom about S1, we get □=1/3≤ α_0 (e.g. α_0 =0.5). It means that the probability of the failure of VC_1 is low. VC_1 will be seen as a spurious fault and abandoned. The threshold α_0 will be set based on the history data and the thresholds in the virtual network and substrate network may be different.

After screening all the nodes and links, an associated component f among F should be found for each symptom s. It means that this symptom s can be explained by f. If f does not exist, the symptom s will be regarded as a spurious symptom and removed from S_o.

B. Maximum probable explanation algorithm

In this section, a heuristic algorithm based on the contribution [2] is proposed to infer the faults from filtered faults which can explain S_o. The contribution of a component c is defined as follows:

$$\text{con (c)} = \sum_{s\in S_{o_c}} \beta(c|s) \ / \sum_{s\in S_c} \beta(c|s) \qquad (2)$$

In addition,

$$\beta(c|s) = p(s|c)p(c) \Big/ \sum_{c\in C_s} p(s|c)p(c) \qquad (3)$$

$p(c)$ means the probability of a component c breaks down. $p(s|c)$ indicates the probability of a symptom s occurs when the component c fails. S_{o_c} is a set of observed symptoms that can be explained by c. S_c is a set of symptoms that should be observed, which can also be explained by c. C_s is a set of components that can explain the symptom s.

Contribution con (c) indicates the likelihood of c can be considered as a fault. In this algorithm, step1-4 initializes all the variables, computes and ranks the contribution of each component. Step 5-14 show that the component with biggest contribution are selected. If it can explain some symptoms in the S_r, it will be inputted into the fault hypothesis H and its relevant symptoms will be removed from S_r. Or if it cannot explain some symptoms in the S_r, it will be abandoned and another component will be chose. This process will be repeated until the set S_r is empty. This make sure that all the observed symptoms can be explained by the components in the set H. Finally, the set H is the fault set that we needed.

The pseudo code of the hypothesis selection algorithm is presented in Algorithm 2:

Algorithm 2: Hypothesis Selection Algorithm
Input: observed symptom set S_o, filtered component set F'
Output: fault hypothesis H
01: H= {};
02: S_r=S_o; // uncovered symptom set
03: $S' = \{\}$; //covered symptom set
04: Q=Sort (F'); // orders all the components from the largest contribution to the smallest;
05: **while** $S_r \neq \emptyset \&\& Q \neq \emptyset$
06: get the first component f from Q;
07: find the symptom set **S**, its symptom can be explained by the component f;
08: **if** $\mathbf{S'} \cap \mathbf{S} \neq \mathbf{S}$
09: remove the common symptoms s from S_r; // s= $\mathbf{S} \cap S_r$
10: add f into the hypothesis H;
11: $\mathbf{S'}$=$\mathbf{S} \cup \mathbf{S'}$;
12: **end if**
13: remove f from Q;
14: **end while**
15: **return** H;

Algorithm 2: Hypothesis selection algorithm

IV. EVALUATION

In this section, we present our experimental environment, evaluation metrics and simulation results.

A. Experimental environment

- **Network topology:** In order to observe the performance of our algorithm under different types of topology models, we use BRITE [11] to create three types of topology model: AS-Waxman, AS-BA and Hierarchical. For each type of topology model, we build five substrate networks and vary network size from 100 to 500. Five virtual networks are also built and their sizes are from 20 to 100.
- **End-to-end observations:** Three nodes are randomly selected as probe stations [12] in the virtual network. The paths consisting of probe stations and nodes follow the shortest path algorithm. Each path is seen as an end-to-end service and the status of path are seen as an observed symptom.
- **Injecting Faults and Noisy observations:** We make each substrate component to break down independently with its prior probability in this experiment. The virtual nodes which mapped to the malfunctioning substrate nodes also break down. In addition, virtual nodes may break down independently. The prior probability of a faulty component is extremely low in the real network environment [13]. Therefore, we randomly set each substrate component with

prior fault probability, which is with Normal distribution (0.005, 0.003). The prior probability of a virtual fault is higher than that of substrate fault. It is with Normal distribution (0.006, 0.003). All the service symptoms, network nodes and links follow the QMR-DT model [13]. The conditional probability between symptoms and faults in the approach MFDM are calculated according to formula (5):

$$p(s=0|f_1,f_2,f_3,\dots)=(1-p_{inhibit})\prod_j p_{leak}^{j} \quad (5)$$

Where, p_{leak}^{j} (Leak probability) means the probability of a probe succeeds when nodes and links that constitute the probe have failed. $p_{inhibit}$ (Inhibition probability) implies the probabilities of a probe failed even when all of nodes and links constitute the probe are normal. In addition, j means the number of failed nodes and links.

B. Metrics

We use two metrics accuracy and false-positive to estimate the quality of this method proposed in this paper. They are defined as follows:

$$\text{Accuracy}=\frac{|H\cap F|}{|F|}, False-positive=\frac{|H\cap\bar{F}|}{|F|}$$

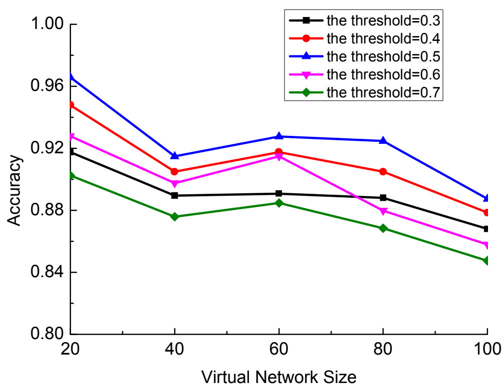
Where H denotes a set of faults which are diagnosed by the system, F means a set of real faults which occurred in the system.

C. Evaluation results

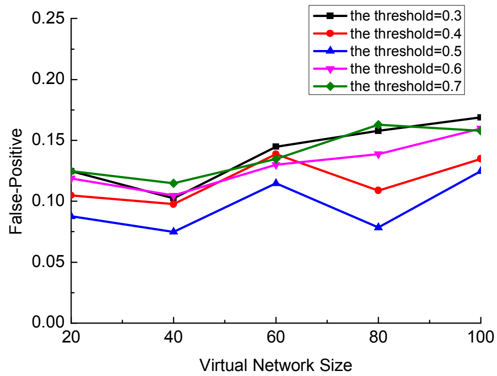
We evaluate the performance of the proposed method from the following four aspects.

1) The impact of the threshold α_0

In order to study the impact of filtering threshold, we run our diagnosis method in both virtual and substrate networks respectively with different threshold and network size. As shown in Fig.2, the accuracy with the threshold α_0=0.5 is higher than others in the virtual network. And the false-positive rate with the threshold α_0=0.5 is lower than others. Similarly, as shown in Fig.3, the accuracy with the threshold α_0=0.5 is also higher than others in the substrate network. And the false-positive rate with the threshold α_0=0.5 is lower than others. So the threshold α_0 in the virtual and substrate network both are set 0.5.

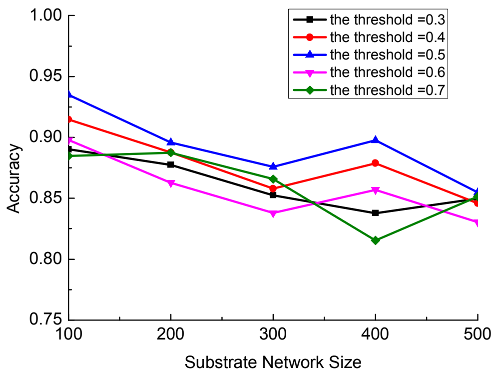


(a) accuracy with different thresholds in the virtual network

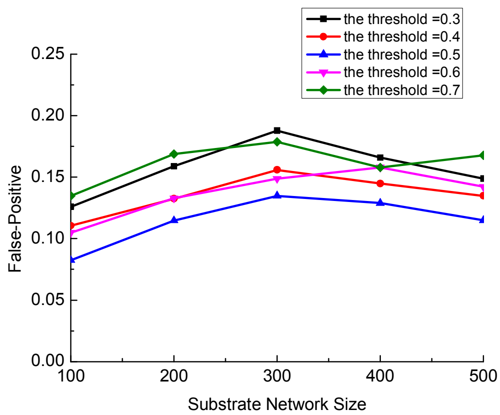


(b) fault positive with different thresholds in the virtual network

Fig.2. **(a)** accuracy with different thresholds in the virtual network**, (b)** false positive with different thresholds in the virtual network



(a) accuracy with different thresholds in the substrate network

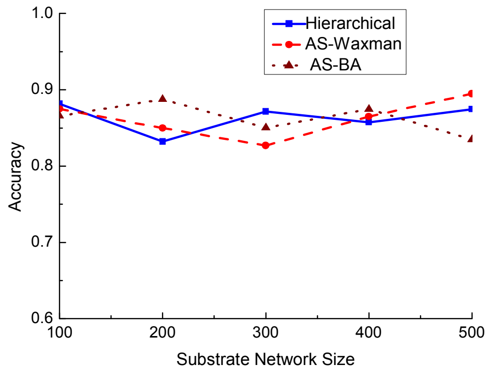


(b) false positive with different thresholds in the substrate network

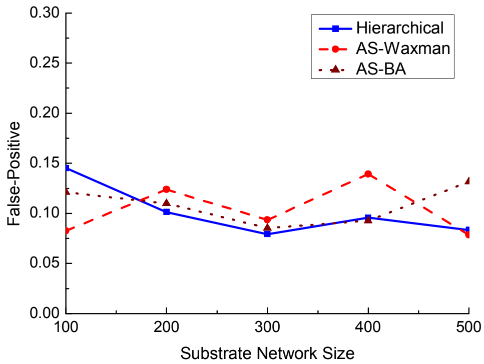
Fig.3. **(a)** accuracy with different thresholds in the substrate network, **(b)** false positive with different thresholds in the substrate network

2) The impact of network topology and size

In order to study the topological effects in our algorithm, three types of topologies are generated by using BRITE: (1) AS-Waxman; (2) AS-BA; (3) Hierarchical. The simulation results in Fig.4 shows that with the increase of substrate network size, the accuracy fluctuated in the range of 0.8-0.9 and the false-positive rate is in the range of 0.05-0.15. The changes of accuracy and false-positive in the different substrate network size are small. In addition, the accuracy rates of three topologies are high and in the same order of magnitude. The false-positive rates of three topologies are low and also in the same order of magnitude. It means that the network topology and size have little effect on our method.



(a) the accuracy in the different types of topologies

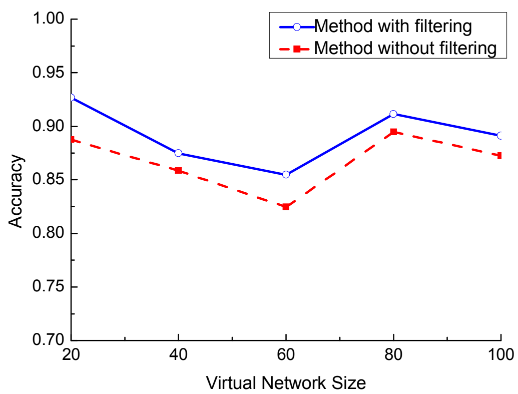


(b) fault positive in the different types of topologies

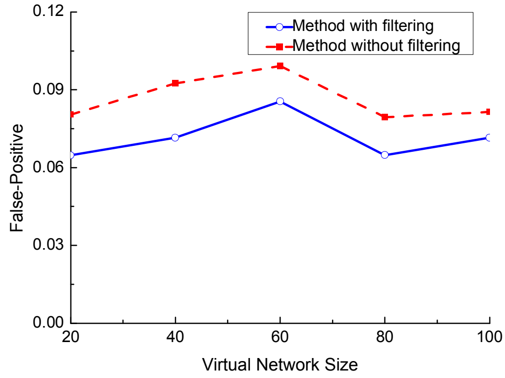
Fig.4. **(a)** accuracy in the different types of topologies, **(b)** fault positive in the different types of topologies

3) The effectiveness of filtering algorithm

Some symptoms may not be received and may be changed into the spurious symptoms in the noisy and dynamic environment. In order to study the effectiveness of filtering algorithm on our method, we run the two methods in the NVE: diagnosis method with filtering and diagnosis method without filtering. As shown in Fig.5, we find that the inference results with the filtering algorithm are better. Its average accuracy rate is 0.04 higher than the diagnosis method without filtering. Its average false-positive rate is 0.05 lower than that with the diagnosis method without filtering. It indicates that our filtering algorithm play a positive role in the diagnosis process.



(a) accuracy in the two diagnostic methods

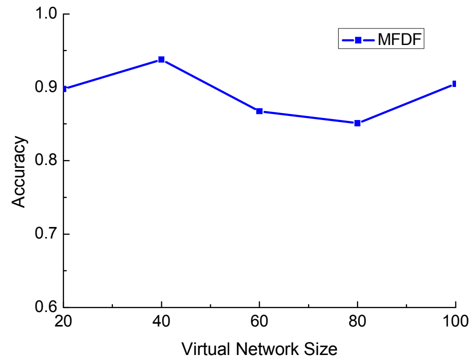


(b) fault positive in the two diagnostic methods

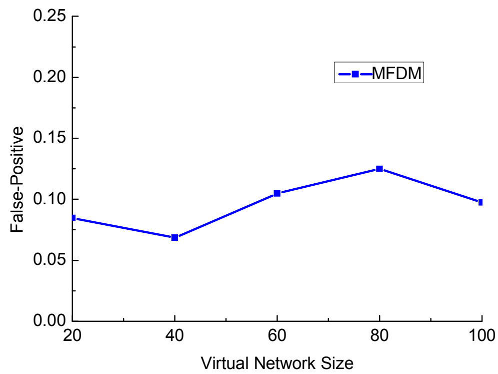
Fig.5. **(a)** accuracy in the two diagnostic methods**, (b)** fault positive in the two diagnostic methods

4) The diagnostic effects of IVFs

Compare with the previous methods, our method can infer the IVFs in the virtual network. In order to validate the diagnostic effects of IVFs in our approaches, we compute the accuracy and false-positive of IVFs. As shown in Fig.6, the accuracy of IVFs is in the range of 0.85~0.95 and the false-positive of IVFs is in the range of 0.05~0.15.



(a)accuracy of IVFs in the virtual network



(b) fault positive of IVFs in the virtual network

Fig.6. **(a)** accuracy of IVFs in the virtual network; **(b)** fault positive of IVFs in the virtual network

V. CONCLUSIONS AND FUTURE WORK

In this paper, we provide a multi-layer fault diagnosis method for the network virtualization environment. It involves two steps: the first is to diagnose all the VFs based on the relationship between the observed symptoms and VFs. The second step is to distinguish IVFs from CVFs and infer the SFs based on the mapping relationship between CVFs and SFs. The results of the experiments show that our fault diagnosis approach can solve the problem which is caused by the inaccessible substrate network information. The filtering mechanism in our method can distinguish CVFs from IVFs, which can improve the quality of fault diagnosis and provide convenience for the fault recovery.

REFERENCES

[1]. Y.L. Pan, X.S. Qiu and S.L Zhang. "*Fault diagnosis in network virtualization environment*." In 18th International conference on telecommunications (ICT), Ayia, 2011: 517-522.

[2]. S.L. Zhang, X.S. Qiu, L.M. Meng. "*Service fault diagnosis algorithm in network virtualization environment.*" Journal of software, 2012

[3]. Y.N. Tang, E. Al-shaer and K. Joshi. "*Reasoning under uncertainty for overlay fault diagnosis*." IEEE Transactions on Network and Service Management,2012: 34-47

[4]. W.G. Chen, L.M. Liang, L. Lin, J. Liu, Y.F. Yue and J.B. Zhao. "*A fault diagnosis method for power transformers based on wavelet neural network and D-S Evidence Theory.*" High Voltage Engineering and Application, Chongqing, 2008: 666-671.

[5]. Malgorzata Steinder, Adarshpal S.Sethi."*Multi-layer fault localization using probabilistic inference in bipartite dependency graphs.*" Technical Report Feb,CIS Dept,2001.

[6]. Richard and E. Neapolitan. "*learning Bayesian netowrks*".

[7]. L. Cheng, X.S. Qiu, L.M. Meng and Y. Qiao, "*Efficient active probing for fault diagnosis in large scale and noisy networks*." In Proc. IEEE INFORCOME,2010.

[8]. M. Steinder, and A. S. Sethi."*Probabilistic Fault Diagnosis in Communication Systems Through Incremental Hypothesis Updating.*" Compute Networks, 2004: 537-562.

[9]. M. Steinder and A. S. Sethi, "*Increasing robustness of fault localization through analysis of lost, spurious and positive symptoums*." INFOCOM, New York,2002:322-331.

[10]. M. Steinder and A. S. Sethi, "*A survey of fault localization techniques in computer networks*." Science of Computer Programming, Special Edition on Topics in System Administration, 2004: 165-194.

[11]. Alberto Medina, Anukool Lakhina, Ibrahim Matta, John Byers. Brite. http://www.cs.bu.edu/brite/. 2013.

[12]. Y. Bejerano, Rajeev Rastogi. "*Robust monitoring of link delays and faults in IP networks.*" IEEE/ACM Transactions on Networking, 2006: 1092-1103.

[13]. T. Jaakkola and M. Jordan, " *Variational probabilistic inference and the qmr-dt database.*" Journal of Artificial Intelligence Research,1999: 291-322.