International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

A Novel Approach for Fault Detection in Computer Networks

Miss. Bhoomika K. Sharma

Department of Computer Engineering
Government Polytechnic, Ahmedabad, Gujarat, India

Abstract:
Over the past few years, fault detection for computer networks has attracted extensive attentions for its importance in management of computer network. Most of the existing fault detection methods are based on active probing techniques. Active probing techniques can detect the occurrence of faults fast and precisely. Especially in large scale networks these methods suffer from the limitation of traffic overhead. A new fault detection method is proposed in this paper to relieve traffic overhead induced by active probing based methods. In this method key is to divide the detection process into multiple stages. Only a small region of the network is detected by using a small set of probes, during each stage. It also makes sure that entire network can be covered after multiple detection stages. This method can give guarantee that the traffic used by probes during each detection stage is small sufficiently so that the network can operate without any disturbance from probes. Few results verify the effectiveness of the proposed method in this paper.

Keywords: Computer networks, Active probing, Probe selection algorithm, Fault detection, Bipartite Bayesian network

1. Introduction

Computer networks play an important role in modern industries. As many other systems, networks might suffer from component failures induced by environment factors or device ageing. Hence, efficient network fault diagnosis methods are needed urgently. Fault diagnosis is the technology that can guarantee the reliability and survivability of systems. During the past few decades, many researchers have considered the network failure problem [1] and [2]. And numerous fault diagnosis methods have also been widely used in the modern industrial systems [3], [4], [5] and [6]. Applying these techniques into computer networks is rather important both theoretically and practically.

For computer networks, fault diagnosis procedure includes two steps: fault detection and fault localization [7]. Fault detection is to identify whether any fault has happened. In this step, detection tools are adopted to detect the presence of network faults. If there exist faults, then fault localization is triggered to identify the fault reason and location. Hence, fault detection is the first key step to perform fault diagnosis for computer networks. To ensure networks' normal operation, quick and accurate fault detection methods are needed.

Classical fault detection tools are usually passive methods. These methods use the alarms sent by the failure components to monitor the operational state of the network and detect the presence of faults [8]. After the faults are detected, they will be further localized based on fault information [9], [10] and [11]. Because of their non-intrusive characteristics, passive detection methods have been widely used [12], [13] and [14]. The main limitation of passive methods lies in their requirement the support of network components [15]. Those passive methods require that each network component should send out an alarm when it fails. In real applications, however, many network devices have no such a function, and the alarms may also be interrupted or lost during transmission. Meanwhile, the

complexity of modern networks could make the relationships between fault alarms and real faults complicated, so that it is difficult to identify fault locations correctly [10] and [16].

In recent years, active probing methods have received many attentions [7], [17], [18], [19] and [20]. These methods detect network faults by sending data packets called probes into networks and inferring the health condition of networks according to the probing results [21]. Thus, they have the advantages of little support from network components and capability to infer network performance quickly and accurately [22]. Active probing detection methods should consume networks' bandwidth resources, so most researchers have focused on probe scheduling methods to minimize the bandwidth consuming [7], [17] and [18]. Meanwhile, some other researchers also propose active probing techniques to solve network black holes problem and the all-optical network fault diagnosis problem [19] and [20].

Previous studies on fault detection based on active probing focus on optimal algorithms that select a set of probes to test the whole network [7] and [17]. When the network is small or with simple topology, these methods can work efficiently; but as the network getting large and complex, these methods may be computationally infeasible [23], and will consume too much network bandwidth. To overcome this shortcoming of existing methods, Barford et al. [23] and Zhao et al. [24] proposed some different strategies to monitor network operation. Barford et al. adopted a simple algorithm to select a set of probes to cover a part of the network each time, which could reduce the traffic overhead at each detection stage and cover all nodes after several rounds of detections. Zhao et al. scheduled path measurements in multiple rounds and monitored a part of the network links in each round. These two approaches provided a new way to fault detection in large scale networks.

Nevertheless, these two approaches require the complete information situation about networks, which means that the relationships between nodes and probes are supposed to be deterministic. In real applications, however, the information about networks is not deterministic because networks can be interfered by many factors [25]. For example, the probe may be influenced by the environment noise. Thus, using complete information assumption to model networks is unsuitable for some cases.

Inspired by Barford's and Zhao's studies, a new fault detection method for computer networks in non-deterministic environment is proposed in this paper. The main idea is to divide the whole detection process into a series of stages. At each stage, a small number of probes are chosen to check some network nodes. After several detection stages, all nodes in the network could be covered. Compared with the existing methods, this method can guarantee that the traffic used by probes during each detection stage is sufficiently small, though more time may be needed to cover the whole network. As shown by simulation results, the number of probes needed at each stage is small, which ensures that the normal operation of networks will not be interfered by detection probes.

The rest of the paper is organized as follows. In section 2, the network model considered in this paper is introduced. Section 3 formulates the problem of fault detection. Section 4 proposes the probe selection algorithm for fault detection. Section 5 presents the simulation results and some discussions. Finally, section concludes this paper.

In this paper, we mainly consider the network nodes' failures. The principle of the active probing method is shown in Fig. 1. In this figure, the network is composed of several network nodes and a network component called probe station. The probe station sends special data packets, which are called probes, to the destination nodes to detect the faults in network. And each probe can be distinguished according to its destination node. According to the probing results, the status of the nodes on the probes' transmission paths can be figured out. Because the effectiveness of this method strongly depends on the normal operation of the probe station and the destination nodes, we assume these network components are always good.

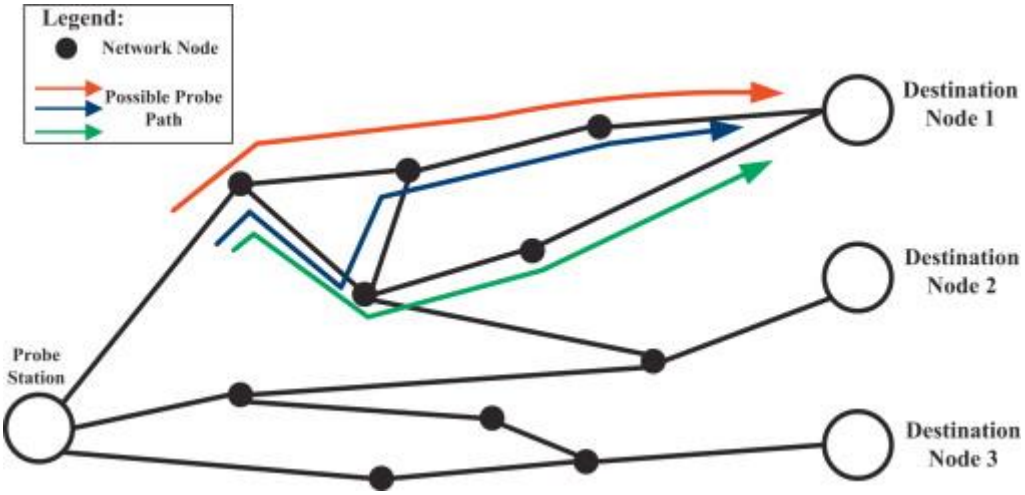

Fig. 1. Schematic diagram of the active probing method.

As active probing technique uses probes to detect network nodes' faults, the relationships between probes and nodes should be considered first. Pervious fault detection researchers mainly use deterministic dependency information, which assumes that the relationships between nodes and probes are completely understood, to model the network [26] and [27]. But in the real networks, the route between two nodes and the network topology may change when the network is operating, which means routes and logical topology are actually uncertain [28]. As shown in Fig. 1, when the probe station sending a probe to the destination node 1, the nodes used to transmit the probe are not deterministic. Because each path from probe station to the destination node 1 could be selected as the transmission path. Meanwhile, the probing results may also be interrupted or lost during detection. Thus, using the deterministic dependency information to model the networks is unsuitable in most cases.

To characterize the non-determinacy of the considered networks, the widely used bipartite Bayesian network [15], [17] and [29] is adopted in this paper to model the probabilistic relationship between nodes and probes. Bayesian network is a powerful tool for modeling and analyzing complex systems. It could be used to describe the complex dependency relationships between random variables [30]. Recently, it has been widely used in reliability engineering and fault diagnosis [21], [31] and [32]. The bipartite Bayesian network is a simplified Bayesian network, which can be used to describe the relationships between nodes and probes. As shown in Fig. 2, a bipartite Bayesian network has a two-layer structure. The upper layer vertexes denote a set of network nodes $N = \{n_i,\ 1 \le i \le K\}$. Each node ni is assigned with a number of 1 to indicate the failure of corresponding node or 0 otherwise. The lower layer represents a set of available probes $P = \{p_j,\ 1 \le j \le M\}$. In this model, if the probe pj fails, then $pj = 1$; otherwise, $pj = 0$. Due to the uncertain characteristics of the networks, the relationships between the nodes and probes are difficult to determine. An efficient way to represent them is to use probabilistic language. In this model, a directed edge between a node and a probe is used to represent the dependency between them. Each edge is associated with a conditional probability $P(pj|ni)$, which denotes the dependency strength between the node ni and the probe pj. The conditional probability $P(pj|ni)$ is the probability of the event that the probe pj fails given that the node ni has failed. The value of $P(pj|ni)$ is mainly determined by the network routing situation, and another two probabilities, leak probability and link probability [33]. Leak probability represents the probability of the event “a probe fails when all nodes it passes are OK”; and link probability is the probability of the event “a probe successes when there are nodes failed on its path”. Hence, the value of $P(pj|ni)$ can be calculated from these probabilities.

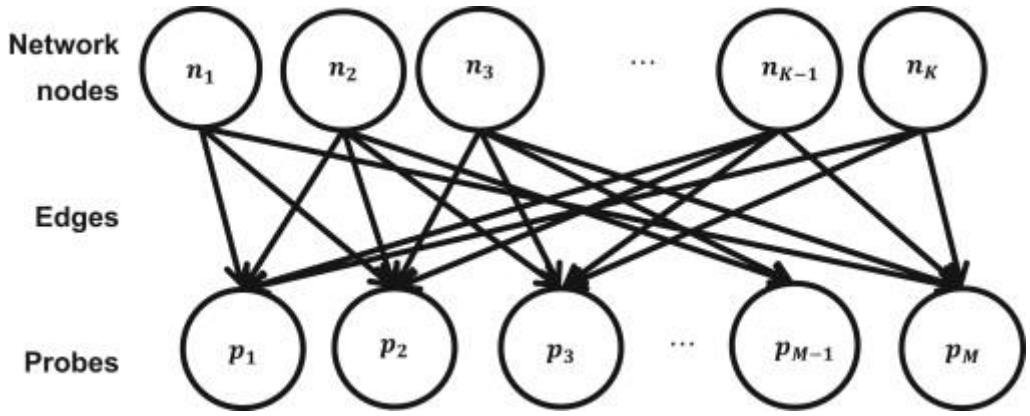

Fig. 2. Bipartite Bayesian network model.

In real applications, the bipartite Bayesian network model can be formulated from the network topological model. The network's topological model gives the information about the network's topology, and mainly describes the connection between the network's nodes and links. When a probe is sent from the probe station to the destination, the nodes that the probe may pass can be learnt from the topological model. Then, combining the network routing information, leak probability and link probability, the relationships between nodes and probes can be represented clearly using probabilistic language, which forms the bipartite Bayesian network model for the network. Hence, no matter how complex a network is, it can be abstracted as the bipartite Bayesian network model according to the topological model and the information about the network's operation conditions.

3. Problem description

Active probing is an efficient network detection method. As mentioned in Section 1, however, existing active probing methods may result in heavy computation and traffic overhead when the network is large. Thus, an efficient but economical detection method for large scale networks is required.

Inspired by Barford's and Zhao's work [23] and [24], a feasible way to avoid heavy computation and traffic over resulted from probes is to divide a detection process into several detection stages. At each stage, a small set of probes are chosen to detect part of the network. Then, the whole network is detected after several detection rounds. Based on this idea and the network's model, the fault detection problem can be described as follows: how to choose a set of probes at each detection stage to monitor part of the network; and how to guarantee the whole network can be covered after a number of detection stages?

As the relationships between the nodes and probes are uncertain, the first problem that needs to be solved is to determine whether a network node is detected. In this paper, a probabilistic description is adopted to describe each node's detected condition. First, the concept Coverage [7] is adopted to quantify the probability of the event that a node is detected. For the node ni at the qth detection stage, $Coverage_q(n_i)$ denotes the probability of the event that at least one probe in all detection probes used fails when node nifails. This metric can be calculated as follows:

$$Coverage_q(n_i)=Coverage_{q-1}(n_i)+Coverage(n_i|P_{det})-Coverage_{q-1}(n_i)\square Coverage(n_i|P_{det}). \quad (1)$$

In this equation, Coverage(ni|Pdet) denotes the Coverage of the node ni when Pdet⊂P is chosen as the detection probe set at the qth stage. It can be calculated by

$$\begin{aligned} Coverage(n_i|P_{det}) &= P(\exists p_j \in P_{det}, p_j = 1 | n_i = 1) \\ &= 1 - P(\forall p_j \in P_{det}, p_j = 0 | n_i = 1) \\ &= 1 - \prod_{p_j \in P_{det}} P(p_j = 0 | n_i = 1) \\ &= 1 - \prod_{p_j \in P_{det}} [1 - P(p_j = 1 | n_i = 1)]. \end{aligned} \quad (2)$$

To check whether a node is detected, a threshold is set for Coverage. If the Coverage of the node ni exceeds the threshold, ni will be considered as detected. A suitable threshold can make the detection accurate and efficient.
Existing active probing algorithms try to cover the whole network at one time. Thus, we can obtain an upper bound for the threshold, and its upper bound is the minimal Coverage of all network nodes when all available probes are chosen as the detection probes. It can be calculated by the following equation:

$$\begin{aligned} threshold_{up} &= \min[Coverage(n_i|P)] \\ &= \min\left\{1 - \prod_{j=1}^{M} [1 - P(p_j = 1 | n_i = 1)]\right\}. \end{aligned} \quad (3)$$

If the value of the chosen threshold is larger than thresholdup, there will be at least one node that cannot be detected even all available probes are chosen.
Nevertheless, there is no need to set an upper bound for the threshold in our proposed method, because we do not try to detect all nodes at one time, and the Coverage of the nodes can accumulate with the detection process. Theoretically, each node's Coverage can exceed any threshold smaller than 1 after several detection stages.
Though an upper bound of the threshold does not need to be set for the proposed algorithm, the value of the threshold could still affect the speed and the accuracy of the detection. When a small threshold is selected, the Coverage of each node could reach the threshold easily, then the detection process can be completed quickly; while the actual probability of each node being detected is small, and the accuracy of the detection is low. On the other hand, selecting a big threshold will make the detection process long and improve the detection accuracy. Thus, the tradeoff between detection speed and accuracy should be made when the threshold is determined.

4. Detection probe selection algorithm

In this section, an efficient algorithm is proposed to choose detection probes at each stage and guarantee that the whole network can be covered after a number of stages. There are three main steps in this algorithm. First, a group of nodes are selected as the target nodes. Second, a set of probes, which have the maximum detection ability, are chosen to detect the target nodes. Last, after several detection stages, all nodes can be detected with a large probability.

In the first step of the proposed probe selection algorithm, the nodes with the highest detection difficulty are chosen as target nodes. Before detection the selected nodes, we should determine the metric that can quantify detection difficulty of a node. In this paper, the smallest but sufficient number of probes needed to detect the node ni, Ki, is adopted as the metric. If a node is with a larger Ki, then more probes are needed to detect the status of this node, which means that this node is more difficult to detect. Thus, using Ki as metric can efficiently quantify the detection difficultly of each node. Given a node ni and the related probes pi1,pi2,…,pik, Ki can be got by the following method. First, the probe with strongest relation to ni is chosen into the detection probe set and the value of Coverage is calculated. Second, the current Coverage is compared to the threshold. If Coverage is larger than the threshold, the number of probes used is Ki. Otherwise, the next probe, whose relation to ni is weaker than the last probe but stronger than remainder unselected probes, is chosen into the probe set. This process last

until Coverage value exceeds the threshold or all available probes are chosen in the probe set. The number of probes used is Ki. The algorithm about the calculation of Ki is shown as Algorithm 1.

Algorithm 1. Ki Calculation algorithm.

Input: bipartite Bayesian network
Output: K_i
Initialize K_i=0, $Coverage(n_i)=0$ for each node $n_i \in N$
for all node $n_i \in N$ **do**
AvailableProbes=*P*
while $Coverage(n_i)<threshold$ **do**
Select the probe *p* with largest $P(p_j \mid n_i)$ from *AvailableProbes*
Remove *p* from *AvailableProbes*
Calculate node's *Coverage*:
$Coverage(n_i)=Coverage(n_i)+P(p_j \mid n_i)-Coverage(n_i) \square P(p_j \mid n_i)$
$K_i=K_i+1$
end while
end for

Algorithm 2. Detection probe selection algorithm.

Input: bipartite Bayesian network model

Output: detection probe set P_{det}

Initialize K_i, $Coverage(n_i)=0$, for each node n_i; *Undetectednodes*=*N*;

while $|Undetectednodes|>0$ **do**

s=1; *AvailableProbes*=*P*; {Initialize}

if $|Undetectednodes|\geq S$ **then**

Select S undetected nodes n with largest K_i into target nodes set N_{tar}

else

Select all undetected nodes n into N_{tar}

end if {formulate target node set}

while $s\leq|N_{tar}|$ **do**

if $n_{i_s}\in N_{tar}$ is undetected **then**

for all probes $p\in P$ are available and $P(p|n_{i_s})>0$ **do**

Calculate each related probe p's *DetectionAbility*

end for {calculate related probes' detection ability}

Choose the probe $p\in AvailableProbes$ with highest *DetectionAbility* into P_{det}

Remove probe p from *AvailableProbes*

Calculate each related probe p's *DetectionAbility*

end for {calculate related probes' detection ability}

Choose the probe $p\in AvailableProbes$ with highest *DetectionAbility* into P_{det}

Remove probe p from *AvailableProbes*

for all $n_{tar}\in N_{tar}$ calculate *Coverage* by **do**

$$Coverage(n_{tar})=Coverage(n_{tar})+P(p|n_{tar})-Coverage(n_{tar})\square P(p|n_{tar})$$

end for

label the target nodes whose $Coverage(n_{tar})$ is bigger than threshold as covered

if n_{i_s} is detected or all related probes are chosen **then**

$$s=s+1$$

end if

else

$s=s+1$ {if node n_{i_s} is detected, go to the next target node}

end if

end while {the detection probe set selection process is over}

update the *Coverage* for nodes $n\in Undetectednodes$

label the nodes whose *Coverage* is bigger than threshold as detected

remove detected nodes from *Undetectednodes*

end while

After getting each node's detection difficulty, the probe selection algorithm starts with selecting the target nodes. The nodes with larger Kis are more difficult to detect, so S nodes with the largest Kis are chosen from the undetected node set, Undetectednodes, into the target node set, Ntar. With the target nodes determined, the algorithm can choose the corresponding probes to detect these nodes one by one. For the undetected target node, nis∈Ntar,1≤s≤S, the related probes with the highest detection ability are chosen as detection probes. Each probe's detection ability at the qth stage can be quantified by the sum of all nodesCoverages when this probe is selected as the detection probe. This metric can be calculated as follows:

$$DetectionAbility_q(p_j) = \sum_{i=1}^{K} Coverage_q(n_i|p_j) \quad (4)$$
$$= \sum_{i=1}^{K} [Coverage_{q-1}(n_i) + P(p_j|n_i) - Coverage_{q-1}(n_i) * P(p_j|n_i)].$$

In this equation, Coverageq(ni||pj) denotes the Coverage of the node ni if it is detected by pj at the qth detection stage. Each time, the algorithm chooses the probe with the largest DetectionAbility into the detection probe set. That is a kind of greedy search strategy which tries to use a small number of probes to detect as many nodes as possible. The object of doing this is to reduce the number of probes used during each stage of the detection process, so the traffic overhead.

After the target nodes are determined, the Coverages of all target nodes are updated. If the Coverage of a node is greater than the threshold, the node is considered as detected. If the target node nis is detected or all related probes are selected, the algorithm will go on to deal with the next undetected target node. Otherwise, it continues adding probes into the detection probe set. The algorithm repeats this process until all target nodes are dealt with, then sends all detection probes into the network. According to the probes used and the network model, the algorithm updates all Coverages of the nodes. The nodes whose Coverages exceed the threshold are considered as detected and removed from the undetected nodes set. If there are any nodes still undetected, the algorithm will return to the target nodes choosing step and repeat this process until all nodes detected. After the whole network is detected, the system resets all nodes Coverages and starts a new detection process. The probe selection algorithm is shown as Algorithm 2.

5. Simulation studies

5.1. Simulation setup

In this section, several simulation experiments are constructed to verify the effectiveness of the proposed network detection method. In these simulations, the bipartite Bayesian network is used as the network model and the algorithm proposed is adopted to detect the network faults.

Theoretically, given a network, the detection system should analyze the topology of the network first; then decides the location of the probe station; finally, it can figure out the relationships between nodes and probes and set up the bipartite Bayesian network model. The probe station selection problem is out of this paper's consideration. For simplicity, this paper gives a network with K nodes and M candidate probes. For each node, some probes are randomly connected to it and the conditional probability relationships between them are generated in random, and uniformly distributed in the range of [0.5,1]. This simulation setting may be too simple for the real networks, but it is sufficient for verifying the effectiveness of our detection method.

5.2. Simulations with different network sizes

In the simulations, a set of networks whose nodes vary from 20 to 200 are considered. The number of available probes M varies from 50 to 500. The threshold of Coverage is 0.95. At each detection stage, the algorithm selects five nodes as the target nodes. For each simulation, we mainly study the number of probes needed at each stage, the total number of used probes, and the number of the detection stages needed. The simulation results are shown in Table 1.

Nodes number K	20	50	70	100	150	200
Probes number M	50	100	150	200	400	500
Detection stages	2	4	8	10	14	18
Total probes	16	34	50	70	111	145
Probes used each stage	13	14	11	14	15	15
	3	11	7	8	12	13
		5	7	9	7	11
		4	9	8	9	12
			7	8	9	11
			5	6	9	10
			3	4	8	6
			1	5	5	7
				5	5	7
				3	9	8
					6	7
					6	8
					5	6
					6	7
						6
						6
						4
						1

Table 1.Result for different network sizes simulation.

The simulation results verify the effectiveness of the proposed method. All nodes can be detected in a limited number of detection stages. At each detection stage, the number of probes is not very large, which can ensure the stability of network operation. On the other hand, the probes used at each stage are with the downward trend, which results from the fact that the nodes chosen at the beginning are more difficult to detect, so more probes are needed. With the detection going on, the target nodes are easier to detect, so the probes needed at each stage become fewer.

5.3. Simulations with different values of S

In this detection method, the number of target nodes, S, at each detection stage is an important parameter that needs further consideration. It has a great effect on the performance of the proposed algorithm, so this simulation is designed to evaluate the effect of S on the algorithm. It uses a network with 50 nodes and 100 available probes as the detection object and repeats the algorithm six times with all parameters same as the simulation in 5.2 except for S. The value of S varies from 1 to 50. The simulation results are shown in Table 2.

Network with 50 nodes and 100 available probes						
S	1	2	5	10	20	50
Detection stages	21	10	5	3	2	1
Total probes	36	34	33	34	35	37
Probes used each stage	3	5	11	18	32	37
	2	5	8	13	3	
	3	2	8	3		
	2	5	5			
	1	3	1			
	1	3				
	2	5				
	2	3				
	2	2				
	1	1				
	1					
	2					
	1					
	3					
	2					
	2					
	2					
	1					
	1					
	1					
	1					
	1					

Table 2. Result for different values of S simulation.

The simulation results show that the value of S may affect the performance of the algorithm greatly. If a small value for S is chosen, the number of probes needed at each stage will be small, and the detection time will be long. While a bigger value is chosen, more probes are needed at each time and the detection stages become fewer. The value of S determines the number of the target nodes. Hence, if more target nodes are chosen each time, more probes are needed, and the detection probes can also cover more nodes each time, so the total detection time is less.

It can also be noticed that the total number of probes needed is not liner with the value of S. When the S is either too small or too big, the algorithm will choose some probes repeatedly, which increase the total number of used probes.

From the above results, we can see that the value of S should be chosen according to the requirement for the detection speed and the detection stability. If fast detection is required, then the large S should be chosen, so that more nodes can be detected at each detection stage. If the stability requirement is put firstly, the small S is required to reduce the influence on the normal network operation.

5.4. Comparison with existing works

This simulation compares the proposed detection method with Chu's method [17]. A simulation with a network having 200 nodes is proposed, and it is also assumed that all parameters are the same as the simulation in 5.2. This method and Chu's method are compared. Chu's method's object is to detect all nodes at one detection stage, and the threshold must satisfy the upper bound constraint. We assume that the threshold in this simulation satisfies this constraint. This simulation mainly focuses on the number of probes used at each detection stage and the number of stages needed to cover the whole network. The simulation results are shown in Fig. 3.

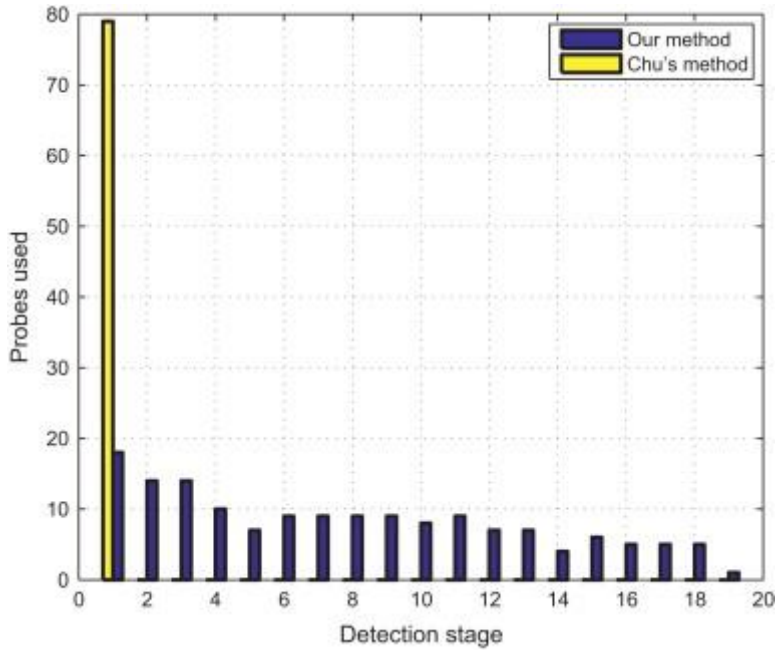

Fig. 3. Probes needed at each stage for two different methods.

As shown in the simulation results, Chu's method can detect all nodes in one detection stage while large number of probes are used at one time. That means a lot of probes will be sent out simultaneously when the network is detected. And the total number of the probes used in that stage is nearly 80. In that situation, the detection process is short, but the normal operation of the network may be affected by the heavy traffic greatly. On the other hand, the method of this paper only sends a small number of probes out each time, which uses no more than 20 probes in each stage. But it needs 19 detection stages to detect the whole network. Although the detection speed of this method is slower than that of Chu's, the detection process is smoother and the affection on the network is smaller. Hence, our proposed method can save the bandwidth resource and guarantee the stability of the network.

5.5. Discussions on simulations

Considering overall performance of the proposed detection method, it can be concluded that this method can check the networks with a limited number of detection stages and probes by choosing the threshold and the number of target nodes properly. The value of S can affect the performance of the algorithm greatly. Choosing a proper threshold can make the detection accurate and efficient. On the other hand, compared with the existing detection methods, the proposed method may use more probes and time to cover the whole network, but the traffic overhead at each stage is reduced significantly. That ensures the stable operation of the network. So we think the proposed method is superior than the existing one in large scale network.

In real applications, once the probe station is determined, the probes that can be used are also determined. As the network's environment is uncertain, the key point of this method is to determine the conditional probability P(pj||ni).

As the conditional probability is affected by the network's environment, such as network routing, network leak probability, and link probability, the accuracy of these information is related to the effectiveness of the proposed method. The routing conditions and the two probabilities can be obtained roughly during the network's operation, which can be used by this detection method. Nevertheless, the methods for obtaining the accurate information should be further studied.

6. Conclusions and future work

In this paper, a fault detection method is proposed for computer networks. Different from the existing studies on network fault detection, this method uses a new active probing strategy to detect faults in non-deterministic network. The main idea of the proposed method is to detect the part of the considered network each time and cover the whole network after several detection stages. This paper describes an algorithm for selecting the detection probes at each detection stage. It can make the fault detection process efficient through reducing traffic overhead of each detection stage. The effectiveness of this algorithm is also verified through a set of simulations.

In this work, the detection method mainly considers detecting persistent faults in networks. In real applications, however, transient faults may happen at any time. If a node fails after it has been detected and returns to the normal condition before it is detected again, the proposed method will not be able to detect this kind of fault. Thus, it is necessary to consider this problem further. Additionally, we also intend to further study the strategies for target nodes selection. In this paper, the target nodes are chosen only based on the detection difficulty. In fact, there are many other factors are worth to take into account, such as node failure probability, node function, and node position. Including these factors can make the detection algorithm more effective and suitable for the computer networks. These problems can be studied in the future.

REFERENCES:

1. C. Rocco, J.E. Ramirez-Marque, Vulnerability metrics and analysis for communities in complex networks Reliability Engineering and System Safety, 96 (10) (2011), pp. 1360–1366
2. C. Zhang, J.E. Ramirez-Marquez, C. Rocco, A new holistic method for reliability performance assessment and critical components detection in complex networks ,IIE Transactions, 43 (9) (2011), pp. 661–675
3. T. Assaf, J.B. Dugan, Diagnosis based on reliability analysis using monitors and sensors, Reliability Engineering and System Safety, 93 (4) (2008), pp. 509–521
4. L.M. Bartlett, E.E. Hurdle, E.M. Kelly, Integrated system fault diagnostics utilising digraph and fault tree-based approaches, Reliability Engineering and System Safety, 94 (6) (2009), pp. 1107–1115
5. Yiannis Papadopoulos, Model-based system monitoring and diagnosis of failures using statecharts and fault trees, Reliability Engineering and System Safety, 81 (3) (2003), pp. 325–341
6. Kyungmee O. Kim, Ming J. Zuo, Two fault classification methods for large systems when available data are limitedReliability Engineering and System Safety, 92 (5) (2007), pp. 585–592
7. Maitreya Natu, Adarshpal S. Sethi, Probabilistic fault diagnosis using adaptive probing ,Lecture Notes in Computer Science, 4875 (2007), pp. 38–49
8. G. Jakobson, M. Weissman, Alarm correlation , IEEE Network, 7 (6) (1993), pp. 52–59
9. Mohamed Abduljalil, Basir Otman. A new probing scheme for fault detection and identification. In: IEEE international conference on electro/information technology. Windsor, ON; June 2009. p. 90–5.
10. M. Steinder, Adarshpal S. Sethi, A survey of fault localization techniques in computer networks, Science of Computer Programming, 52 (2) (2004), pp. 165–194
11. Mohamed Abduljalil, Basir Otman. Fusion based approach for distributed alarm correlation in computer networks. In: Second international conference on communication software and networks. Singapore; February 2010. p. 318–24.

12. Su Ming-Shan, Thulasiraman K, Das A. A scalable on-line multilevel distributed network fault detection/monitoring system based on the SNMP protocol. In: IEEE global telecommunications conference. Taipei, Taiwan, vol. 2; November 2002. p. 1960–4.
13. Rouvellou I, Hart G. Automatic alarm correlation for fault identification. In: Proceedings of the IEEE INFOCOM '95, vol. 2. Boston, MA; April 1995. p. 553–61.
14. Tang Yongning, Al-Shaer Ehab. Overlay fault diagnosis based on evidential reasoning. In: Proceedings of the IEEE INFOCOM. Rio de Janeiro, Brazil; April 2009. p. 2596–600.
15. Cheng Lu, Qiu Xuesong, Meng Luoming, Qiao Yan, Boutaba Raouf. Efficient active probing for fault diagnosis in large scale and noisy networks. In: Proceedings of the IEEE INFOCOM. San Diego, CA; March 2010. p. 1–9.
16. Nancy Samman, Ahmed Karmouch, Network anomaly diagnosis via statistical analysis and evidential reasoning, IEEE Transactions on Network and Service Management, 5 (2) (2008), pp. 65–77
17. Chu LW, Zou SH, Chen SD. Wang WD, Tian CQ. Internet service fault management using active probing in uncertain and noisy environment. In: 4th international conference on communications and networks in China. Xian, China; August 2009. p. 1–5.
18. Nguyen Hung X, Teixeira Renata, Thiran Patrick, Diot Christophe. Minimizing probing cost for detecting interface failures: algorithms and scalability analysis. In: Proceedings of the IEEE INFOCOM. Rio de Janeiro, Brazil; April 2009. p. 1386–94.
19. Harvey Nicholas JA, Patrascu Miai, Wen Yonggang, Yekhanin Sergey, Chan Vincent WS. Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs. In: Proceedings of the IEEE INFOCOM. Anchorage, AK; May 2007. p. 697–705.
20. Kompella Ramana Rao, Yates Jennifer, Greenberg Albert, Snoeren Alex C. Detection and localization of network black holes. In: Proceedings of the IEEE INFOCOM. Anchorage, AK; May 2007. p. 2180–8.
21. Mark Brodie, Irina Rish, Sheng Ma, Intelligent probing: a cost-effective approach to fault diagnosis in computer networks, IBM System Journal, 41 (3) (2002), pp. 372–385
22. Jeswani Deepak, Korde Nakul, Patil Dinesh, Natu Maitreya, Augustine John. Probe station selection algorithms for fault management in computer networks. In: Second international conference on communication systems and networks (COMSNETS). Bangalore, Indian; January 2010. p. 1–5.
23. Barfod Paul, Duffield Nick, Ron Amos, Sommers Joel. Network performance anomaly detection and localization. In: Proceedings of the IEEE INFOCOM. Rio de Janeiro, Brazil; April 2009. p. 1377–85.
24. Zhao Yao, Zhu Zhaosheng, Chen Yan, Pei Dan, Wang Jia. Towards efficient large-scale VPN monitoring and diagnosis under operational constraints. In: Proceedings of the IEEE INFOCOM. Rio de Janeiro, Brazil; April 2009. p. 531–9.
25. Li Cheng, Zou Shihong, Chu Lingwei. Online learning based internet service fault diagnosis using active probing. In: International conference on networking, sensing and control. Okayama, Japan; March 2009. p. 773–8.
26. Maitreya Natu, Adarshpal S. Sethi, Errol L. Lloyd, Efficient probe selection algorithms for fault diagnosis, Telecommunication Systems, 37 (1–3) (2008), pp. 109–125
27. Natu M, Sethi Adarshpal S. Efficient probing techniques for fault diagnosis. In: Second international conference on internet monitoring and protection. San Jose, CA; July 2007. p. 20–5.
28. M. Steinder, Adarshpal S. Sethi, Probabilistic fault localization in communication systems using belief networks, IEEE/ACM transactions on networking, 12 (5) (2004), pp. 809–822
29. O. Doguc, J.E. Ramirez-Marquez, An automated method for estimating reliability of grid systems using Bayesian networks Reliability Engineering and System Safety, 104 (1) (2012), pp. 96–105
30. H. Boudali, J.B. Dugan, A discrete-time Bayesian network reliability modeling and analysis framework, Reliability Engineering and System Safety, 87 (3) (2005), pp. 337–349

31. Ozge Doguc, Jose Emmanuel Ramirez-Marquez, A generic method for estimating system reliability using Bayesian networks, Reliability Engineering and System Safety, 94 (2) (2009), pp. 542–550
32. Alyson G. Wilson, Aparna V. Huzurbazar, Bayesian networks for multilevel system reliability, Reliability Engineering and System Safety, 92 (10) (2007), pp. 1413–1420
33. Rish Irina, Brodie Mark, Odintsova Natalia, Ma Sheng, Grabarnik Genady. Real-time problem determination in distributed systems using active probing. In: IEEE/IFIP network operations and management symposium. Seoul, South Korea; April 2004. p. 133–46.