Fault Diagnosis for High-Level Applications Based on Dynamic Bayesian Network*

Zhiqing Li, Lu Cheng, Xue-song Qiu, and Li Wu

Networking and Switching Technology State Key Laboratory,
Beijing University of Posts and Telecommunications, Beijing 100876
zhiqing323@bupt.cn

Abstract. In order to improve the quality of Internet service, it is important to quickly and accurately diagnose the root fault from the observed symptoms and knowledge. Because of the dynamical changes in service system which are caused by many factors such as dynamic routing and link congestion, the dependence between the observed symptoms and the root faults becomes more complex and uncertain, especially in noisy environment. Therefore, the performance of fault localization based on static Bayesian network (BN) degrades. This paper establishes a fault diagnosis technique based on dynamic Bayesian network (DBN),which can deal with the system dynamics and noise. Moreover, our algorithm has taken several measures to reduce the algorithm complexity in order to run efficiently in large-scale networks. We implement simulation and compare our algorithm with our former algorithm based on BN (ITFD) in accuracy, efficiency and time. The results show that our algorithm can be effectively used to diagnose the root fault in high-level applications.

Keywords: fault diagnosis, dynamic Bayesian network, approximate inference, noisy environment.

1 Introduction

In order to ensure the availability and effectiveness of the network and improve the quality of services, it is important to quickly and accurately diagnose the root fault from the observed symptoms and knowledge. Because High-level applications are implemented in distributed systems, any fault in software or hardware would probably cause problems of availability or quality of services. Anyway the more complex of the service, the more uncertain of the relationship between symptoms and root faults [1]. Therefore in order to diagnose these faults, some probabilistic inference techniques are present, such as techniques based on Bayesian networks.

At present, people pay much attention on probabilistic fault diagnosis based on Bayesian networks which mostly include IHU [2], Shrink [3], Maxcoverage [6] and so on. All these algorithms assume that the state of the managed system does not

* **Foundation Items:** Sub-topics of 973 project of China (2007CB310703), Fok Ying Tung Education Foundation(111069).

C.S. Hong et al. (Eds.): APNOMS 2009, LNCS 5787, pp. 61–70, 2009.

change during the time interval of fault diagnosis. They can complete fault localization and perform relatively well using the current observations. However, they do not consider the changes of node status during the time of observing and inference. In distributed systems, routing and states of one component between end-to-end services may change dynamically caused by link congestion, dynamic routing or other reasons. If the network is large, the observation period must be set long, therefore the status of the same node in one observation period may change, we call these nodes contradiction nodes which are incorrectly considered to be caused by noises in the algorithms based on Bayesian networks. This increases the possibility of diagnostic errors.

Historical faults in management information base reflect the information of the system state through which we can analysis and get the probabilistic message and the state transition message of each node. If there is one node which fails more frequently, the prior probability of this node should be updated and become larger. However in BN the network is assumed to be static, hence the prior probability (characteristic) of each node is assumed to be fixed. So fault diagnosis techniques based on dynamic Bayesian networks attract our attention.

1.1 Related Work

Dynamic Bayesian network (DBN) [4] extends static Bayesian network (BN) by introducing the notion of time, namely, by adding time slices and specifying transition probabilities between these slices. DBN use the Markov assumption that the future system state is independent of its past states given the present state. As a result, a DBN is defined as a two-slice BN, where the intra-slice dependencies are described by a static BN, and inter-slice dependencies describe the transition probabilities. This feature just describes the dynamic systems. It is for this reason that diagnosis techniques based on dynamic Bayesian network become more and more important in probabilistic fault diagnosis techniques for dynamic system. The latest research results about exact inference include forwards-backwards algorithm, frontier algorithm and the interface algorithm [4] etc. Because the computational complexity of exact algorithms based on DBN is so high that we will lose diagnostic significance. Then we convert to approximate algorithms which include The Boyen-Koller (BK) algorithm [5], the factored frontier (FF) algorithm [4] and PF algorithm [7]. BK algorithm takes in a factored prior, does one step of exact updating using the jtree algorithm, and then projects the results back down to factored form. Unfortunately, for some models, even one step of exact updating is intractable. FF algorithm and PF algorithm has a lower diagnosis accuracy although their computational complexity is not so high.

1.2 Contributions in This Paper

In this paper, we propose a new probabilistic inference technique based on DBN, which is able to localize root causes in noisy and dynamic environments with high accuracy and efficiency. The observed symptoms could be probing results by active measurement [8] [9], or alarms which are received by the management station. Based on the symptoms, the algorithm can locate faults of high-level applications in dynamic system and show

that it has a better performance than the inference based on static Bayesian model. Because characteristic of each node in dynamic system changes influenced by network environments, we add time information to DBN model to handle this dynamics and propose a simple tool to deal with noise. Moreover, our algorithm has taken several measures to reduce the algorithm complexity in order to diagnose in large-scale networks.
The contributions made by this paper are:

1) We use DBN model to deal with system dynamics. In other words our algorithm uses transition probabilities between time slices to update the prior probability of the nodes which can improve the accuracy greatly.
2) We propose an approximate algorithm to reduce the computational complexity in large-scale networks. First, in adjacent time slices, we only update the probability of the nodes whose statuses are faulty, and the transition probability of normal nodes is approximately considered to be $P(F^t=0|F^{t-1}=0)=1$. Thus, if one node fails more frequently, the prior probability of it increases more rapidly. Second, if the observations between adjacent time slices are same, we hold that the state of each node does not change, so we do not perform inference and then we output the results diagnosed in the former time slice.
3) We implement the algorithm based on DBN and ITFD[10] and then compare them in accuracy, efficiency and time at 0% noise level and 5% noise level. Experiments show that the algorithm based on DBN outperforms that based on BN.

2 DBN Model for Diagnosis

As DBN is an extension of BN in time slices, first of all, we construct the static BN (Fig.1. (a)) using EM algorithm according to symptom nodes, fault nodes and the dependences between them. The BN model is a two layer directed graph, and then we extend the model by introducing the notion of time. Each node has two states: 0(normal) or 1(faulty). For a fault F_i, if it happens we denoted $F_i=1$ and $F_i=0$ otherwise. The general inference problem in Bayesian networks is NP hard, and the running time is exponential with the number of nodes. If the probability $P(|F|>k)$ is very small, ignoring such low probability events would make a good trade-off between increasing the running time and lowering the accuracy.

The DBN model is as Fig.1.(b). The first time slice is the initial part, each fault node has a prior $P(F_i^1)$, and each symptom node has a conditional probability table (CPT) $P(S_i^1|F_i^1)$ where S_i^1 is i'th symptom node at time 1 and F_i^1 is i'th fault node at time 1. The second part is a two-slice temporal Bayesian net (2TBN), for each fault node, it also has a transition probability $P(F_i^t|F_i^{t-1})$ which reflects the transfer characteristics of each node as the time slice goes. The value of it also depends on the characteristics of the network. For example, if the visits of one component is too large, it is more likely to be faulty in the next time slice, so the value of both $P(F_i^t=1|F_i^{t-1}=1)$ and $P(F_i^t=1|F_i^{t-1}=0)$ are larger than $P(F_i^t=0|F_i^{t-1}=0)$ and $P(F_i^t=0|F_i^{t-1}=1)$.Our algorithm aims to calculate the posterior probability $P(F^t|F^{t-1},S^t)$, where F^t is the set of root causes, S^t is a set of observations. Our algorithm performs fault diagnosis based on this model.

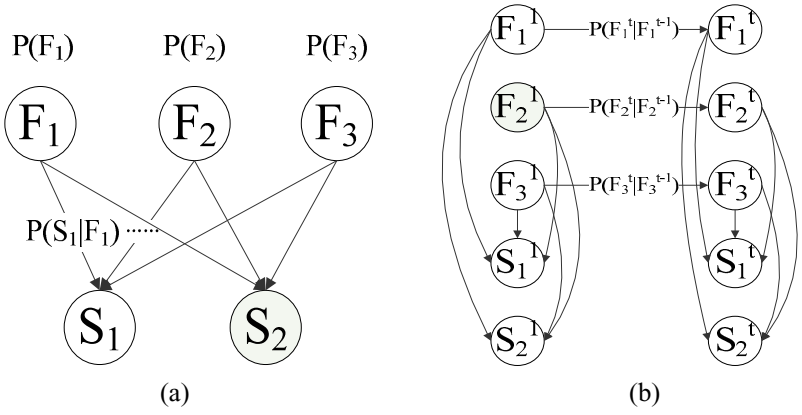

Fig. 1. (a) BN with 5 nodes (b) DBN with 5 nodes by extending BN in time slice. The model is a two-slice DBN (B_1, B_t), where B_1 is for initial time slice and B_t for each subsequent time slice, we perform inference literately in B_t. In the intra-slice both B_1 and B_t are static BN, in the inter-slice, we perform approximate inference by only updating the nodes with state changing.

3 Fault Localization Using DBN Model

In this section, we will introduce our approach for fault localization in High-level applications using dynamic Bayesian network.

The fault nodes of DBN correspond to the logical links or physical links. The probability of each fault node which reflects the property of the node changes with the dynamic network environments. The symptom nodes correspond to the end-to-end services or some applications. When failures take place in some of these nodes in one time slice, malfunction or decrease of quality of services will be observed. The prior probability of these nodes should be updated and increased. Therefore if the failure of one node takes place more frequently, the probability of this node should become larger, which is great important for us to diagnose the root fault. For example, if the database denies to be accessed, the root can be that the host which the database belongs to is down. It also can be that some of the links congest or the packages are lost. All these reasons which are corresponding to the fault nodes in DBN happen independently with a probability. If the bandwidth of network is relatively small, link congestion may often take place in some time slices. For this reason, the probability of nodes corresponding to link congestion should be updated more than others and become larger.

It can be seen that the characteristic of each node in DBN will change with the dynamic network environments. However in BN the network are assumed to be static, hence the probability(characteristic) of each node are assumed to be fixed and does not change. In the realistic application, if the failure of one node takes place more frequently, the probability of this node should become larger. It is very important for us to diagnose the root and improve the performance of the algorithm.

3.1 Algorithm Description

We utilize active probing method to detect the failures. In order to simulate the dynamics of High-level applications, when injecting faults, we produce the faults according to the prior probability in the first time slice but in the subsequent time slices we produce faults according to the transition probability and the state of this node in the last time slice. Then we produce symptoms and send them to the diagnosis engine.When the diagnosis engine receives the suspected nodes F_{sus}^{t} and observations S_o^t, first we determine the time slice information.

If it is in the first time slice, we use an observation rate $Ratio_{Fi}$ for each node to filter the spurious faults and observations. After filtering spurious nodes which are mostly caused by noise, we obtain a new set of suspected nodes which will be used to filter the whole DBN model. In other words in DBN model we only reserve the symptoms which can be explained by the fault nodes in the new set of suspected nodes. Then we start the diagnosis algorithm based on BN. During the inference, we obtain one or more sets of nodes $H_i=\{F_i^t=1,F_j^t=1\ldots\ldots F_k^t=1\}$ called Hypothesis, each of which can explain all the symptoms independently. For each Hypothesis H_i, we calculate a brief $B(H_i^t,S_o^t)$ as the equation (1),where S_o^t is the observations, $pa(S_j^t)$ is a set of nodes which can explain S_j^t, F^t is the set of all fault nodes in the whole DBN model. We obtain the most likely Hypothesis H^{*t} as equation (2) and output H^{*t} as results of the algorithm. Meanwhile, the posterior probability of each node in H^{*t} should be calculated as equation (3). Finally we update the prior probability for each node in the next time slice in H^{*t} using the posterior probability and transition probability as equation (4).

$$B(H_i^t, S_o^t) = \prod_{F_i^t \in F^t} P(F_i^t) \prod_{S_j^t \in S^t} P(S_j^t \mid pa(S_j^t))$$

$$= \prod_{F_i^t \in H_i^t} P(F_i^t = 1) \prod_{F_i^t \in F^t / H_i^t} P(F_i^t = 0) \prod_{S_j^t \in S_o^t} P(S_j^t = 1 \mid pa(S_j^t)) \prod_{S_j^t \in S^t / S_o^t} P(S_j^t = 0 \mid pa(S_j^t)) \quad (1)$$

$$H^{*t} = \arg\max_{H_i^t} B(H_i^t \mid S_o^t) = \arg\max_{H_i^t} B(H_i^t, S_o^t) \quad (2)$$

$$Postprior(F_i^t \mid pa(F_i^t)) = p(F_i^t = 1) \prod_{F_j^t \in F^t / F_i^t} p(F_j^t = 0) \prod_{s_j^t \in s_o^t} p(s_j^t = 1 \mid F_i^t = 1) \quad (3)$$

$$p(F_i^t) = \sum_{F_i^{t-1}} p(F_i^t \mid F_i^{t-1}) * Postprior(F_i^{t-1} \mid pa(F_i^{t-1})) \quad (4)$$

If it is not in the first time slice, we first compare the symptoms in the current time slice S_o^t with the symptoms in the last time slice S_o^{t-1}. If they are exactly the same($S_o^t=S_o^{t-1}$),we believe that none of nodes have changed, that is the state of the system does not change. Therefore we output the most likely Hypothesis H^{*t-1} in the last time slice as the results of the current time slice. If the symptoms are not the same ($S_o^t!=S_o^{t-1}$), we get every node $F_i^{t-1} \in H^{*t-1}$ and add it to the set of suspected

nodes in the current time slice $F_{sus}^{t} = F_{sus}^{t} \cup H^{*t-1}$ because the fault nodes in H^{*t-1} may still be faulty in the current time slice. An observation rate $Ratio_{Fi}$ for each node is used to filter spurious nodes and observations caused by noise. In order to reduce the complexity in DBN model we only reserve the symptoms which can be explained by the fault nodes in the new set of suspected nodes. Then the inference based on BN will be started and output one or more Hypothesis H_i^t. We calculate the brief of each Hypothesis $B(H_i^t, S_o^t)$ as equation(1) and output the most likely Hypothesis H^{*t} as equation(2).At the same time the posterior of each node in H^{*t} can be calculated as equation(3). Finally we update the prior probability for each node in the next time slice in H^{*t} using the posterior probability and transition probability as equation (4).The workflow of our algorithm is shown as Fig. 2.

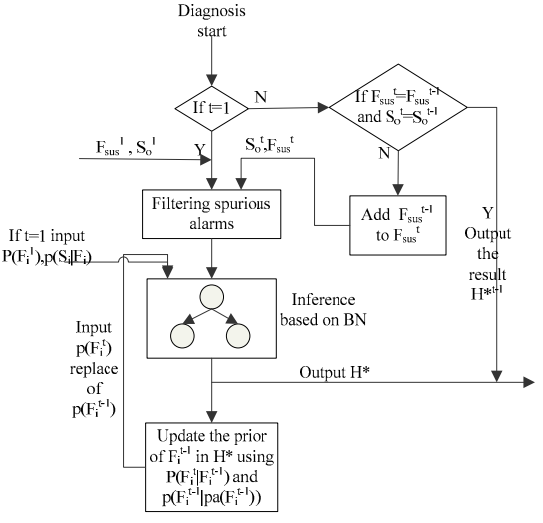

Fig. 2. The workflow of the algorithm based on DBN

In the distributed system, when an end-to-end service failure takes place, each node in its path can cause it probably. So we call these nodes suspected nodes F_{sus}. For each node in F_{sus}, if another end-to-end service which is successful passes through it, we consider this node to be ok and remove it from F_{sus}. If there are no nodes in F_{sus}, we hold that the end-to-end service failures are caused by noise which should be filtered using a mechanism before inference.

We propose a simple but effective tool as equation (5)[11] to deal with spurious faults and symptoms caused by noise. Experiment show that it can work well and improve the accuracy with less running time. For a real fault, it is highly unlikely that all symptoms can be seen. However if the number of observations is lower than a threshold, we consider it to be spurious fault. For example, if a node failure F_i^t can arouse five symptoms, but the observations are only two, suppose all the conditional probability are sane, so $Ratio_{Fi}$=0.4 which is lower than a threshold 0.5. This node

will be considered to be spurious and be removed from F_{sus}^t. If no node in F_{sus}^t can explain a symptom, we hold that this symptom is spurious and remove it from the observations.

$$Ratio_{F_i^t} = \frac{\sum_{S_i \in S_o} P(S_i^t \mid F_i^t)}{\sum_{S_i \in S} P(S_i^t \mid F_i^t)} \quad (5)$$

3.2 Analysis of Complexity

The complexity of algorithms based on DBN at present is so high that it is difficult to diagnose in large-scale networks. Because in each time slice we should perform update for each node and do inference in the whole DBN. Generally, the number of nodes is so large that even performing one time update and inference is intolerable. To reduce the computational complexity, we improve the algorithm as follows:

1) If the observations received in the current time slice are the same as observations in the last time slice ($S_o^t=S_o^{t-1}$), the status of each node will be considered not to change. The diagnosis results in the last time slice will be outputted as results of the current slice. The algorithm only needs to do one time comparison and not to do inference, so the time is $O(|S_o|)$.
2) If the observations are different between adjacent time slice ($S_o^t!=S_o^{t-1}$),we add H^{*t-1} to F_{sus}^t and perform diagnosis because the nodes in H^{*t-1} may be still faulty in the current slice. The algorithm only update the prior probability of the node in H^{*t}, thus if a node becomes faulty more frequently, the prior of it is larger than others, which is great important to improve the accuracy. Here we consider the transition probability $p(x_i^t=0|x_i^{t-1}=0)=1$ approximately.
3) After filtering we only reserve the symptoms which can be explained by the fault nodes in the new set of suspected nodes. The DBN model is simplified so much that the algorithm complexity reduces highly. If the number of real fault is $K(K<|F_{sus}|)$, the maximum number of Hypothesis $|F_{sus}|^k$, so the time is $O(|F_{sus}|^k)$ not $O(|F|^k)$,where $|F|$ is the whole number of fault nodes and $|F_{sus}|<<|F|$.

4 Simulation

In this section, we simulate the network environment and adopt active probing method to obtain the symptoms and detect suspected fault. Then we implement ITFD and our algorithm with java and compare them in accuracy, efficiency and time.

4.1 Simulated Networks

We use network simulation tool INET[11] to generate the network topology. The connection between each pair of nodes is considered as an end-to-end service which corresponds to the symptom node in DBN model whose status could be possibly observed. The faults of DBN correspond to network edges. If a service passes though an

edge, then we add dependence relationship between them. Finally we extend BN by adding time information and obtain DBN model.

For each fault node, we produce a prior probability which can be used to inject fault in the first time slice. For each symptom node, we produce a conditional probability which represents the influence from fault node to symptom node. In the first time slice, for each fault node we randomly generate a number range such as[0,1] referring to prior probability. If the prior probability falls into the range, this node will be considered as faulty and the symptom node which traverses this fault node will be fail as observed. In the subsequent time slice, for each fault node whose status is 0, we randomly generate a number range referring to transition probability $P(F_i^t=1| F_i^{t-1}=0)$, otherwise referring to $P(F_i^t=0| F_i^{t-1}=1)$.If the transition probability fall into the range, the status of this node will be changed from 0 to 1 or from 1 to 0. Because the transition probability $P(F_i^t=1| F_i^{t-1}=1)$and $P(F_i^t=0| F_i^{t-1}=0)$ are close to 1, each node can most probably keep its status in adjacent time slice. For each real fault, all symptom nodes which traverse this fault node will fail as observed.

For each observation, all fault nodes in its path can be added into F_{sus}. For each fault node in F_{sus}, if at least one symptom node whose status is 0 passes through it, this node will be considered as normal and removed from F_{sus}. Both the suspected nodes F_{sus} and observations are sent to diagnosis engine.

In distributed systems, the status of symptom nodes may change probably during propagation such as loss of packet or spurious symptoms. If some observations are lost, the diagnosis engine may feel incorrectly that there is no such observation. If the engine receives some spurious symptoms, it is useless to do such inference. For this reason, we inject spurious symptoms with a noise level 0% or 5%. In other words, we change the status of symptom nodes from 0 to 1 or from 1 to 0 with the noise level.

4.2 Metrics

We utilize two metrics named accuracy and false positive to evaluate the performance of the two algorithms. H is the set of diagnosis results, F is the set of real faults.[2]

$$Accuracy = \frac{H \cap F}{F}, \qquad falsepositive = \frac{H - F}{F}$$

4.3 Simulation Results

We implement our algorithm based on DBN (ADBN)) and the ITFD based on BN, where the time slice in DBN is T=50. All the figures show results averaged 50 runs.

Without noise: Fig.3.(Left) compares the diagnosis accuracy of the algorithm based on DBN and the algorithm based on BN(ITFD). As the number of nodes becomes larger, the accuracy of ADBN remains higher than 90% but that of ITFD declines slightly. When the number of nodes is more than 200, the accuracy of ITFD drops sharply but that of ADBN declines elegantly. With the number of nodes up to 500, the accuracy of ITFD is worthless.The false positive of ADBN is lower than ITFD shown in Fig.3(Right),in other words,the precision of ADBN is higher than ITFD.As number of nodes is more than 200, the precision of ITFD drops sharply but that of ADBN drops slightly.

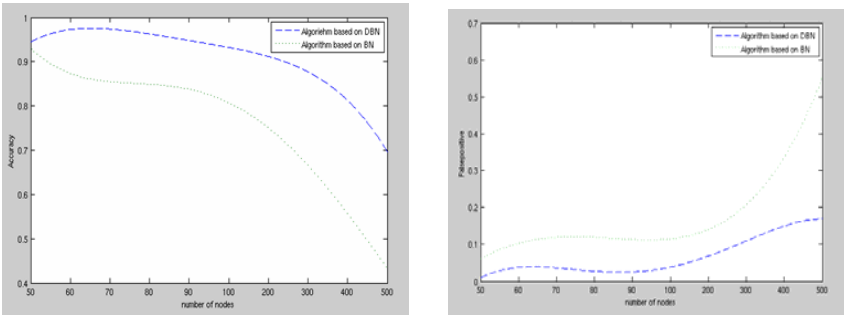

Fig. 3. Comparison of the two algorithms with noise=0% (Left) Accuracy (Right) Precision. The falsepositive of ADBN is lower than that of ITFD,that is the precision of ADBN is higher than that of ABN.

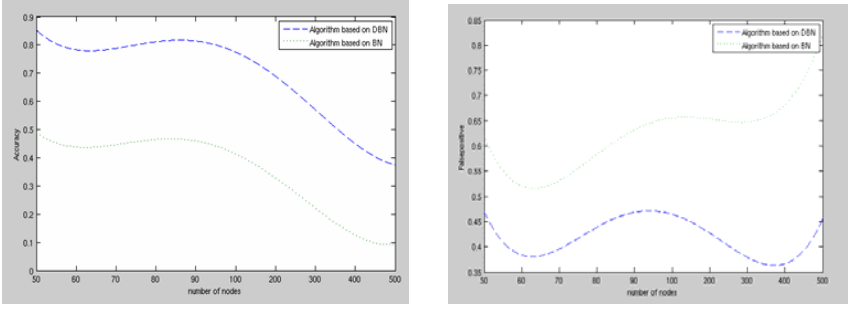

Fig. 4. Comparison of the two algorithms with noise=5% (Left) Accuracy (Right) Precision

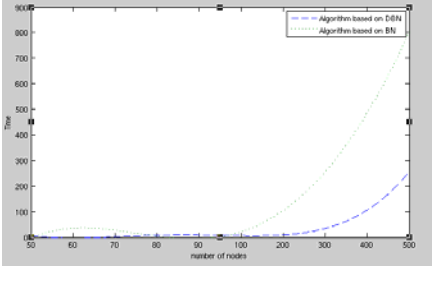

Fig. 5. Comparison of the two algorithms in fault localization time. Both of the two algorithms runs quite rapidly with the number of nodes less than 200. However, as the number of nodes is more than 200, the running time of ITFD increases exponentially but that of ADBN is still less than 300ms.

With 5% noise: As the number of nodes becomes larger, the accuracy of both of the two algorithms declines but ADBN is higher than ITFD by 40% shown in Fig.4.(Left). We can propose that the ITFD cannot work in noise environment. As the

result of the noise, the accuracy is lower than that without noise. The false positive of the two algorithms with 5% noise are higher than that without noise shown in Fig.4.(Right).However the precision of ADBN is still higher than that of ITFD.

Time: Fig.5 shows that the running time of the two algorithms is close with the number of nodes less than 200, both quite rapidly. However, as the number of nodes is more than 200, the time of ITFD increases exponentially but that of ADBN increase slightly. With the number of nodes up to 500, the time of ADBN is still less than 300ms which is quite acceptable in large-scale networks.

5 Conclusion and Future Work

We have presented a technique to diagnose and locate end-to-end services faults in dynamic system and showed that it has a better performance than the inference based on static Bayesian model. We used dynamic Bayesian model to handle system dynamics and a simple tool to deal with noise. Our approximate algorithm has taken several measures to reduce the algorithm complexity in order to diagnose in large-scale networks.

Our new technique can perform fault localization for the current time using the historical alarms and the current observations, but it cannot predict when and which node will fail. We believe that the fault prediction will become more and more important in the management of IP network.

References

1. Steinder, M., Sethi, A.S.: Probabilistic fault localization in communication systems using belief networks. IEEE/ACMTransactions on Networking (in press)
2. Steinder, M., Sethi, A.S.: Non-deterministic event-driven fault diagnosis through incremental hypothesis updating. In: Goldszmidt, G., Schoenwaelder, J. (eds.) Integrated Network Management VIII, Colorado Springs, CO, pp. 635–648 (2003)
3. Kandula, S., Katabi, D., Vasseur, J.P.: Shrink: A tool for failure diagnosis in IP networks. In: Proc. ACM SIGCOMM (2005)
4. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning, PHD degree thesis of University of California, Berkeley (2002)
5. Boyen, X., Koller, D.: Tractable inference for complex stochastic processes. In: Proc. of the Conf. on Uncertainty in AI (1998)
6. Kompella, R.R., Yates, J., Greenberg, A., Snoeren, A.: Detection and Localization of Network Black Holes. In: Proc. IEEE INFOCOM (2007)
7. Arulampalam, S., Maskell, S., Gordon, N.J., Clapp, T.: A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Tran. on Signal Proc. 50(2), 174–188 (2002)
8. Natu, M., Sethi, A.S.: Probe Station Placement for Fault Diagnosis. In: Proc. IEEE GLOBECOM 2007 (2007)
9. Natu, M., Sethi, A.S.: Probabilistic fault diagnosis using adaptive probing. In: Clemm, A., Granville, L.Z., Stadler, R. (eds.) DSOM 2007. LNCS, vol. 4785, pp. 38–49. Springer, Heidelberg (2007)
10. Cheng, L., Qiu, X.-s., Meng, L., Qiao, Y., Li, Z.-q.: Probabilistic Fault diagnosis for IT Services in Noisy and Dynamic Environments. In: Proc. IM 2009 (2009)
11. Winick, J., Jamin, S.: Inet-3.0: Internet topology generator, Technical Report CSE-TR-456-02, University of Michigan (2002)