This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

Efficient Active Probing for Fault Diagnosis in Large Scale and Noisy Networks

Lu Cheng, Xuesong Qiu, Luoming Meng and Yan Qiao
State Key laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications,
Beijing, China, 100876
Email: {chenglu, xsqiu, lmmeng}@bupt.edu.cn

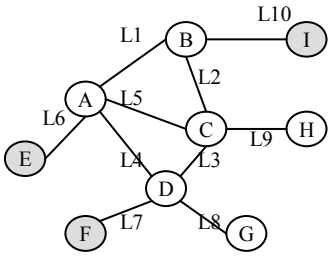
Raouf Boutaba
Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
Email: rboutaba@uwaterloo.ca

***Abstract*- Active probing is an effective tool for monitoring networks. By measuring probing responses, we can perform fault diagnosis actively and efficiently without instrumentation on managed entities. In order to reduce the traffic generated by probing messages and the measurement infrastructure costs, an optimal set of probes is desirable. However, the computational complexity for obtaining such an optimal set is very high. Existing works assume single-fault scenarios, apply only to small size networks, or use simplistic methods that are vulnerable to noises. In this paper, by exploiting the conditionally independent property in Bayesian networks, we prove a theorem on the information provided by a set of probes. Based on this theorem and structure property of Bayesian networks, we propose two approaches which can effectively reduce the computation time. A highly efficient adaptive probing algorithm is then presented. Compared with previous techniques, experiments have shown that our approach is more efficient in selecting an optimal set of probes without degrading diagnosis quality in large scale and noisy networks.**

Index Terms—Fault diagnosis, Bayesian networks, information theory

I. INTRODUCTION

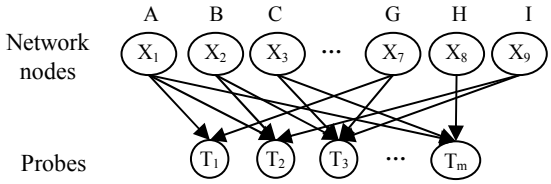
The rapid growth in size and complexity of computer systems and networks make the task of fault management an increasingly important and extremely difficult. Classical approaches to fault diagnosis in network management are mostly based on passive event correlation [1][2]. The typical limitation of these techniques is that they require heavy instrumentation since each managed entity needs to have the ability to send out the appropriate alarms. But in many real applications, only network devices are able to report status information initiated, for example, through SNMP traps. Many managed entities, such as applications developed by third parties, middleware and hosts, do not even provide management interfaces, let alone event notifications that reflect their status. In these cases, the management system has to actively inquire about their health. In recent years, active probing which provides end-to-end monitoring capability has received significant attention [3]-[9]. By measuring probing responses, fault diagnosis can be performed without instrumentation on managed entities. Moreover, fault diagnosis can be accomplished more efficiently if we can actively observe specific data points.



(a) A logical network in that gray nodes are connected directly with probe stations

T_1: E→A→D→G
T_2: F→D→A→B→I
T_3: I→B→C→D→G
……
T_m: E→A→C→H

(b) Examples of Probes



(c) A bipartite bayesian network that models the relationship between network nodes and probes

Fig. 1 **Problem Setup**: Usually there is a large number of candidate probes. When failures are detected, we want to select an optimal set of probes that maximizes the probing quality and minimizes the probing cost.

The Bayesian network is a graph model which has been widely applied in many fields. Similar to previous works[3]-[7], the notion of a bipartite Bayesian network is useful for modeling the relationship between network nodes and probes, as shown in Figure 1. The network nodes may be physical or logical entities, such as routers, servers, applications, or service invocations between applications. A probe T, such as a database query, is a method of obtaining information about these nodes. The response of T depends on the states of the nodes on its path, which is constantly disturbed by noise and inaccurate knowledge of the probing path.

Each vertex in Fig. 1(c) represents a random variable. The upper layer vertex X_i gets value 1 if the corresponding node breaks down and 0 if it is OK. The lower layer vertex T_j gets the value 1 if its corresponding probe fails and 0 otherwise. Each node X_i fails independently with a marginal probability $P(X_i = 1)$. A vertex in the upper layer is connected with directed edges to the lower layer vertices denoting that the failure would lead to negative responses of these probes. An edge from X_i to T_j is weighted by the conditional probability of probe T_j given that X_i has failed, i.e., $P(T_j = 1 | X_i = 1)$. The

978-1-4244-5837-0/10/$26.00 ©2010 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

larger this conditional probability is, the stronger the dependency between these nodes are. Similar to previous works [3]-[7], we assume a noisy-OR model of probability distribution in which several causes contribute independently to the same probe, namely all the causes of a probe are combined using logical operator OR.

In fault management, active probing can be used for two tasks, namely fault detection and fault diagnosis. Fault detection is to discover if there is at least one faulty component in the system. When faults are detected, fault diagnosis is used to identify all the faulty components. Optimal diagnosis probes must be chosen dynamically from a large number of candidates in real-time. The computational complexity of determining such an optimal set is very high. Existing works assume single-fault scenarios, apply only to small scale networks, or use simplistic methods which are vulnerable to noises. In this paper, by exploiting conditionally independent property in Bayesian networks, we present an efficient approach for selecting an optimal set of probes without degrading the diagnosis quality in large scale and noisy networks.

The rest of this paper is organized as follows. We describe the problem of active probing for fault diagnosis in Section 2. In Section 3, we introduce the conditionally independent property in Bayesian networks and prove a theorem. In Section 4, we propose two approaches and a highly efficient probing algorithm. In Section 5, we present our evaluation results. In Section 6, we introduce some related work. Finally, we conclude in Section 7.

II. Active Probing for Fault Diagnosis in Networks

A. Formulation of Active Probing

The main criticism of active probing is it can be costly both in terms of additional traffic due to probing messages and measurement infrastructure cost. Therefore, a small number of probes is usually desirable. Due to limitations in practical applications, choices for sites of probe stations where probes are issued are restricted. Therefore, most works on this topic [3]-[9] have been concerned with selecting an optimal subset of probes from a large number of candidates.

The perfect subset **D** of probes is such that the quality of probing $Q(\boldsymbol{D})$ is maximized, while the cost $C(\boldsymbol{D})$ is minimized. But generally, it is hard to find such an ideal subset. Because, usually when:

$$\boldsymbol{D_1} \subseteq \boldsymbol{D_2}, i.e. C(\boldsymbol{D_1}) \leq C(\boldsymbol{D_2}), we\ have\ Q(\boldsymbol{D_1}) \leq Q(\boldsymbol{D_2}).$$

The most common formulation of this problem is to consider a budget $B \in \mathbb{R}$, and find the optimal set $\boldsymbol{D}^*$ such that:

$$\boldsymbol{D}^* = argmax\ Q(\boldsymbol{D})\ subject\ to\ C(\boldsymbol{D}) \leq B \qquad (1)$$

An alternative formulation is to consider a specified quality of probing $V \in \mathbb{R}$, and try to find the optimal set $\boldsymbol{D}^*$ such that:

$$\boldsymbol{D}^* = argmin\ C(\boldsymbol{D})\ subject\ to\ Q(\boldsymbol{D}) \geq V$$

Considering the context of fault diagnosis for communication networks and systems addressed, in this paper, we will adopt the first formulation. For simplistic, we also assume that probe cost is additive in this paper.

B. Quality Function for Fault Diagnosis

Suppose the state of the system is denoted by a vector $\boldsymbol{X} = (X_1, X_2, \ldots X_n)$ of random variables, where X_i represents the state of a node. In this paper, we use bold-face letters to denote vectors. A probabilistic model is a joint probability distribution (a prior distribution) over all the random variables X_i. Hence, it assigns, a probability P(**X**=$\boldsymbol{x}$) to each realization $\boldsymbol{x}$. Given observations $\boldsymbol{D}$ of the system state, a fault diagnosis algorithm identifies the Most Probable Explanation (*MPE)* of the underlying system responsible for observed outage, i.e.:

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \mathrm{P}(\boldsymbol{X} = \boldsymbol{x}|\boldsymbol{D})$$

Observations reduce uncertainty of the system. Before we observe $\boldsymbol{D}$, the uncertainty about system state can be quantified by Shannon entropy:

$$H(\boldsymbol{X}) = \sum_{\boldsymbol{X}} P(\boldsymbol{X}) log \frac{1}{P(\boldsymbol{X})} = -\sum_{\boldsymbol{X}} P(\boldsymbol{X}) log P(\boldsymbol{X})$$

in that $0\log 1/0 = 0$.

If the system state is doubtless, H($\boldsymbol{X}$) =0. As uncertainty goes up, H($\boldsymbol{X}$) increases. The purpose of making observations is just to reduce this uncertainty. Given probe responses $\boldsymbol{D} = (T_1, T_2, \ldots T_m)$, the posterior probability that the system is in a state changes into *P(**X**|**D**),* consequently uncertainty of the system state is:

$$H(\boldsymbol{X}|\boldsymbol{D}) = \sum_{\boldsymbol{D}} P(\boldsymbol{D}) H(\boldsymbol{X}|\boldsymbol{D}) = -\sum_{\boldsymbol{X},\boldsymbol{D}} P(\boldsymbol{X},\boldsymbol{D}) log P(\boldsymbol{X}|\boldsymbol{D})$$

Noises such as packet loss and dynamic changes, which are the main reasons why fault diagnosis is difficult, lead to opposite observations of the fact, but on average, observations reduce uncertainty, i.e. $H(\mathbf{X}|\mathbf{D}) < H(\mathbf{X})$. In other words, reduction of entropy is a good measure of the information gain of observations for fault diagnosis.

Earlier, Lindley [11] proposed a method to measure the information provided by an experiment. Suppose the probe T is defined on system state $\boldsymbol{X}$:

Definition 1([11]). The average amount of information gain from T, with prior probability distribution of system state P($\boldsymbol{X}$), is:

$$G(T) = H(\boldsymbol{X}) - H(\boldsymbol{X}|T) = \sum_{\boldsymbol{X},T} P(\boldsymbol{X},T) log \frac{P(\boldsymbol{X},T)}{P(\boldsymbol{X})P(\mathrm{T})} \qquad (2)$$

As a consequence and in order to select an optimal set of probes for fault diagnosis, our target is to find a subset $\boldsymbol{D}^*$ which maximizes the reduction of Shannon entropy subject to total cost no bigger than Budget B, i.e.:

$$\boldsymbol{D}^* = argmax\ G(\boldsymbol{D})\ subject\ to\ C(\boldsymbol{D}) \leq B \qquad (3)$$

The problem stated in equation (3) is NP-hard [3], hence efficient exact solutions are likely not possible. Therefore, the heuristic greedy approach, which iteratively adds a probe that reduces the maximum entropy of the system, has been widely used in the literature. Indeed, the greedy algorithm achieves near-optimal performance in many situations.

In [3] and following works such as [4] and [5], Brodie et al. regarded fault diagnosis problem as the task of reducing the uncertainty about the current system state $\boldsymbol{X}$ by acquiring more

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

information from probe T. When problems are detected, a greedy algorithm repeatedly selects the next most-informative probe T* that maximizes the conditional mutual information of $\boldsymbol{X}$ and T, given the previously selected probes $\boldsymbol{D}$, i.e.:

$$T^* = arg\ max_{T \in \boldsymbol{D} \setminus \boldsymbol{D}'} \{H(\boldsymbol{X}|\boldsymbol{D}') - H(\boldsymbol{X}|\boldsymbol{D}', \mathrm{T})\} \quad (4)$$

In (4), not only is the mutual information of **X** and T measured, but the impact of selected probes on T is also calculated, which provides a "global view", better than the techniques which only consider system state **X** and the candidate probe T.

However, equation (4) is an approximation of (3) and its computation complexity is still pretty high. Because even if every node has two states (good or faulty), the total number of system states is up to 2^n. When the network size increases, computation time increases rapidly. This is the main reason why [3]-[5] focus on the approximation of (4) only when all the nodes have the same prior faulty probability in a single-fault scenario. Under this circumstance, the total number of system states reduces to n+1, thus the running time decreases significantly.

In[12], Zheng et al. made a significant improvement on [3]-[5], and, to the best of our knowledge, represents the best solution to the problem in (4). They proposed an approximation that utilized the belief propagation (BP) [13] infrastructure to simultaneously compute approximations of marginal and conditional entropies on multiple subsets of nodes. The method is applied to multiple-fault scenarios in realistic Internet-like topologies. They demonstrated that the average approximation error does not exceed 2%. In addition, it is two orders of magnitude faster than regular greedy methods according to our experiments. However, their technique has two limitations: (1) In the experiments, for a network with 500 nodes and 524 candidate probes, the algorithm requires 37 minutes on average which limits its application in large networks. (2) This work is based on belief propagation, so it has the limitations that BP has, such as for small prior probability the algorithm does not converge. However, the prior probability of a faulty node being on is quite low in real networks [14]. The authors of [15] also pointed out that in realistic environments the probability of a failure is significantly low; the probability a router fails in any given hour is about 10^{-6}.

C. Paper Contributions

Shannon entropy is a good metric of uncertainty of system state, and mutual information measures quite well how observations of random variables (i.e. probes) reduces uncertainty of other variables (i.e. system nodes), or in other words, increases the accuracy of fault diagnosis. Existing techniques assume single-fault scenarios, apply only to small scale networks, or use simplistic methods which are vulnerable to noises. Inspired by previous works, we have made the following contributions in this paper:

(1) Exploiting conditionally independent property in Bayesian networks, we prove a theorem about the mutual information provided by a set of probes.
(2) Based on this theorem and the conditional independence property of Bayesian networks, we propose two approaches which can effectively reduce the computation time, especially in large scale networks. A highly efficient adaptive probing algorithm is then presented.
(3) BPEA[12] which is two orders of magnitude faster than the regular greedy methods, is the best known solution to the problem addressed in this paper. Our experiments show that our approach achieves nearly identical quality of diagnosis, but is one order of magnitude faster than BPEA in networks with 500 nodes.

Furthermore, compared to BPEA is limited to BP, many inference techniques on Bayesian networks can be used to implement our algorithm. This broadens the applicability of our approach to a larger spectrum of management areas such as QoS assurance, performance monitoring, intrusion detection and others.

III A THEOREM ON ACTIVE PROBING EXPERIMENTS

A. Conditional independence in Bayesian networks

In a Bayesian network, if X and Y are connected by an edge, X and Y are dependent and information can be transmitted over this edge. The knowledge of X will influence belief about Y, and vice versa. Otherwise, if X and Y are not connected directly, information is transmitted over the path between them. Here a path is a sequence of non repeated vertices such that from each of its vertices there is an edge to the next vertex in the graph. For example, in Fig 1.(c), a path between T_1 and X_9 is $T_1 \leftarrow X_7 \rightarrow T_3 \leftarrow X_9$.

In a Bayesian network, if all the paths between X and Y are blocked by a set **Z** of nodes, information cannot be transmitted between them. We say that X and Y are directed separated by **Z**, or X and Y are d-separated by **Z** [13]. There are three graph patterns of d-separation, namely serial connection, diverging connection and converging connection [13].

Theorem 1 ([13]). Given a Bayesian network, let X and Y be two variables and **Z** be a set of variables that does not contain X or Y. If **Z** d-separates X and Y, then $X \perp Y \mid \mathbf{Z}$.

For example, if $T_1 \leftarrow X_7 \rightarrow T_3 \leftarrow X_9$ is the only path between T_1 and X_9, this path is blocked when X_7 is observed. In other words, T_1 and X_9 are conditionally independent given X_7.

In the Bayesian network shown in Fig 1.(c), any two probes do not have an edge between them, neither do the network nodes. Therefore, a path between any two probes will include network nodes. In addition, a network node and its neighboring probes form diverging connection, such as $T_1 \leftarrow X_7 \rightarrow T_3$. When the state of a network node is known, all the paths that contain this node will be blocked. Therefore, if we acquire the state of the network, that is to say, if the states of all of the network nodes are determined, any path between any two probe nodes is blocked. Formally, given the state of

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

the network, any two probe nodes in a Bayesian network are independent, i.e.:

$$\mathrm{P}(T_1, T_2, \ldots T_m|\boldsymbol{X}) = \prod_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P}(\mathrm{T_i}|\boldsymbol{X})$$

In the next subsection, we will use this conditionally independent property in Bayesian networks in our proof.

B. Information gain of a set of probes

In this section, we first introduce a theorem proposed by Lindley [11], and prove that it is still true for any finite number of probes.

We have the following important theorem: the experiment T_1 is defined on space $\boldsymbol{X_1}$, and the experiment T_2 is defined on space $\boldsymbol{X_2}$. Here $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ are subspaces of $\boldsymbol{X}$.

Theorem 2 ([11]). If $\mathrm{T_1}$ and $\mathrm{T_2}$ are conditionally independent, i.e. $\mathrm{P}(\mathrm{T_1}, \mathrm{T_2}|\mathbf{X}) = \mathrm{P}(\mathrm{T_1}|\mathbf{X})\mathrm{P}(\mathrm{T_2}|\mathbf{X})$, then: $G(T_2|T_1) \leq G(T_2)$

with equality if, and only if, $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ are independent.

The theorem stipulates that if T_1 and T_2 are independent, either one is more informative, on the average, if performed firs. Particularly, if T_1=T_2, the theorem says that an independent repeat of the same experiment is of less information, than the original experiment.

We can generalize Theorem 2 to cases with any finite number of experiments. We assume experiment T_1 is defined on space $\boldsymbol{X_1}$, experiment T_2 is defined on space $\boldsymbol{X_2}$, … , and T_m is defined on space $\boldsymbol{X_m}$. Here $\boldsymbol{X_1}$, $\boldsymbol{X_2}$, … , $\boldsymbol{X_m}$ are subspaces of $\boldsymbol{X}$.

Theorem 3. For any finite number of conditionally independent experiments $T_1, T_2, \ldots T_m$, i.e. $\mathrm{P}(T_1, T_2, \ldots T_m|\boldsymbol{X}) = \prod_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{P}(\mathrm{T_i}|\boldsymbol{X})$, let $\boldsymbol{D^{(0)}} = \emptyset$, $\boldsymbol{D^{(1)}} = \{T_1\}, \boldsymbol{D^{(2)}} = \{T_1, T_2\}$, ..., $\boldsymbol{D^{(i)}} = \{T_1, T_2, \ldots T_i\}, (1 \leq i \leq m < +\infty)$, then:

$$G(T_j|\boldsymbol{D^{(i)}}) \leq G(T_j|\boldsymbol{D^{(i-1)}})$$

with equality if, and only if, given $\mathbf{D^{(i-1)}}$, $\mathrm{T_i}$ and $\mathrm{T_j}$ are conditionally independent.

This is a property which agrees with the common belief in the diminishing marginal utility of independent observations. The proof is provided in the appendix.

IV. An Improved Adaptive Probing Algorithm

Based on conditional independence in Bayesian networks and Theorem 3, this section proposes two approaches which can effectively reduce computation time, especially in large scale networks. Combined, a highly efficient adaptive probing algorithm is presented.

Similar to existing works, we use greedy approach, which begins with the empty set $\mathbf{D} = \emptyset$, and iteratively adds the probe T which maximally increases the information gain, or in other words, decreases the entropy.

A. Utilizing monotony of information gain

In existing adaptive probing algorithms, every time when it comes to select the next probe, owing to updated selected probes, the information gain of all the remaining candidate probes will be reevaluated and the best one will be selected. Because the total number of system state increases exponentially with the number of network nodes, it is very time-consuming even to reevaluate the information gain of a single probe, especially in large scale networks.

According to Theorem 3, for any candidate probe $\mathrm{T_j}$, its information gain in the *i*th round $G(T_j|\boldsymbol{D^{(i)}})$ will decrease or remain the same compared to that in the last round $G(T_j|\boldsymbol{D^{(i-1)}})$, i.e.: $\alpha = G\left(T_j|\boldsymbol{D^{(i-1)}}\right) - G\left(T_j|\boldsymbol{D^{(i)}}\right) \geq 0$

Please note that the information gain is non-increasing. Hence, for two candidate probes T_l and T_k, if in the last round of evaluation $G\left(T_l|\boldsymbol{D^{(i-1)}}\right) \geq G(T_k|\boldsymbol{D^{(i-1)}})$ and:

$$G\left(T_l|\boldsymbol{D^{(i)}}\right) \geq G\left(T_k|\boldsymbol{D^{(i-1)}}\right) \qquad (5)$$

in this round we need not to reevaluate $\mathrm{T_k}$ and those probes whose information gain are not greater than that of $\mathrm{T_k}$ in the last round. Without evaluation, we already know:
$G\left(T_1|\boldsymbol{D^{(i)}}\right) \geq G\left(T_k|\boldsymbol{D^{(i)}}\right)$ and $G\left(T_1|\boldsymbol{D^{(i)}}\right) \geq G\left(T_o|\boldsymbol{D^{(i)}}\right)$.

Here T_o refers to one of those probes whose information gain is less than that of $\mathrm{T_k}$ in the last round of evaluation.

Further, if $G\left(T_1|\boldsymbol{D^{(i-1)}}\right)$ and $G\left(T_\mathrm{k}|\boldsymbol{D^{(i-1)}}\right)$ are the second largest and third largest gain respectively, when (5) holds, in the *i*th round only T_l has to be reevaluated. There is no need to update the information gain of $\mathrm{T_k}$ and other candidate probes.

In fact, because the system state can be preliminarily evaluated on the basis of probing responses from fault detection, the probability of system state $\mathrm{P}(\mathbf{X})$ will not change significantly if the majority of probing responses reflect correctly the ground truth. Through an empirical study we obseve that, in many occasions, $G\left(T_j|D^{(i-1)}\right) - G\left(T_j|D^{(i)}\right)$ is a small value. That is to say, the number of probes that has to be reevaluated is small most of the time, this way saving a significant amount of time. More specifically, in our experiments, for networks with 500 nodes and 524 candidate probes, the number of probes needed to be evaluated never exceeds 10, usually less than 4 by employing this method.

B. Approximate Conditional Independence of Probes

Even if various causes of uncertainty exist, the negative impact of noise is uniformly distributed in the network. Though spurious symptoms exist, for a good node there often is only a fraction of all its negative symptoms that are seen. For example, as shown in Fig. 1(c), E is connected directly with a probe station. If a great number of probes must traverse E and A to reach other destinations and a majority of these probes are successful, then the probability that E or A is faulty becomes very small. Hence, we can say that the states of E and A are nearly certain (OK), and uncertainty is negligible. Such nodes are often located near probe stations, or are interconnection nodes in the backbone.

We use a simple but effective approach in our previous work [10] to determine the states of network nodes. We use a threshold as follows:

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

$$\beta_{x_i} = \sum_{T_j \in \mathbf{D_o}} P(T_j = t | X_i = x) \Big/ \sum_{T_j \in \mathbf{D_k}} P(T_j = t | X_i = x)$$

Here $\mathbf{D_o}$ means actually observed, $\mathbf{D_k}$ means candidate probes (knowledge). For a network node X_i, if the number of probes which depend on it exceeds k (Here k is related to uncertainty in the network; k grows as uncertainty grows), and $\beta_{x_i} \geq \beta$ (β is a constant value between 0 and 1), then the probability $P(X_i = x)$ is very high. In our experiments, we set k to be 4. Even in situations with 30% noise level, the probability of approximation error is very small.

Based on this and if we consider that the states of such nodes are approximately determined, we show great savings in terms of running time.

From Fig. 1(c) we can see that a network node and its neighboring probes form diverging connection. According to Theorem 1, if the state of a network node is nearly certain, all the paths traversing it will be blocked. If all of the paths between two probes are blocked by such nodes, we say these two probes are conditionally independent.

According to Theorem 3, if T_i and T_j are conditionally independent, and T_i is the probe selected in the ith round, then in the $(i+1)$th round of evaluation, the information gain of T_j will remain the same as the one in the last round, i.e. $G(T_j|\boldsymbol{D}^{(i)}) = G(T_j|\boldsymbol{D}^{(i-1)})$.

Further, if: (1) The probes which are selected from the ith round to $(i+k)$th round are T_i, T_{i+1}, …, T_{i+k}, $(1 \leq k < +\infty)$; (2) Given $\mathbf{D^{(i-1)}}$, a set of network nodes whose states are nearly determined block all of the paths between T_i and T_{i+y} ($\forall y \in [0, k]$), then: $G(T_j|\boldsymbol{D}^{(i-1)}) = G(T_j|\boldsymbol{D}^{(i)}) = \cdots = G(T_j|\boldsymbol{D}^{(i+k)})$.

Intuitively, if two probes have been blocked by some network nodes, because this blockage pattern of diverging connection will never be changed by any new observations of probing responses, T_i and T_{i+y} will remain independent in the (k+1) rounds of evaluation. That is to say, for a probe T_j the information gain changes only when a probe that is dependent on it gets selected, otherwise its information gain will never change.

Consequently, before updating information gain of a probe, if no probe dependent on it gets selected since last evaluation, we don't need to reevaluate it. For example, based on the non-increasing property of information gain, if T_i has the largest information gain in the ith round evaluation and gets selected, T_j has the second largest, and T_i and T_j are independent given $\mathbf{D^{(i-1)}}$, then $G(T_j|\boldsymbol{D}^{(i-1)}) = G(T_j|\boldsymbol{D}^{(i)})$. We can select T_j directly without updating it.

In the experiments, we find that this approximation hardly incurs any loss of information, while in addition to performance promoted by utilizing non-increasing property of information gain, this technique speeds up our algorithm.

C. An Improved Adaptive Probing Algorithm

Combining the two above discussed techniques, we devise an improved adaptive probing algorithm which is much more efficient than previous ones.

Algorithm IAP: Improved Adaptive Probing
Input: Candidate probes ***S***, Budget B, detection results $\mathbf{D_{initial}}$
Output: the most probable system state **Mpe**

```
D= D_initial;
for (Probe T∈S) do Gain(T) = updateGain(T, D); end for
while(S≠{})
   sort(S); //descending sort S in terms of gain
   if ( ! isDependent( getSelected(T_max), T_max)) do
      MaxGain = Gain(T_max);
   else MaxGain = updateGain(T_max, D);
   end if
   i=2; //the first probe in S is T_max
   while (MaxGain < Gain(T_i)) do
      if ( isDependent(getSelected(Ti), Ti) ) do
         MaxGain = Gain(Ti);
      else
         Gain(Ti) = updateGain(Ti, D);
         if (MaxGain<Gain(Ti)) do MaxGain=Gain(Ti); end if
      end if
      i++;
   end while
   if ( cost( D ∪ ( T = arg(MaxGain))) ≤ B ) do
      execute selected Probe T; it returns T = t;
      D = D ∪ {t}; remove T;
   else
      break;
   end if
end while
return Mpe = arg max_x P(X|D)
```

Fig. 2. Improved Adaptive Probing (IAP) Algorithm

There are typically two techniques, namely pre-planned and adaptive, for using heuristic greedy approach to select probes for fault diagnosis. In pre-planned probing, we compute the smallest probe-set such that each system state will produce a different set of probe outcomes, allowing the state to be uniquely determined. In turn, the adaptive method selects the most informative probe at a time, and then waits until the observation of this probe is returned before determining the next probe to be sent. Adaptive probing is an incremental technique that is well-suited for real-time fault diagnosis. Besides, it avoids the waste inherent in the pre-planned approach since it always selects and sends probes as needed in response to problems that have actually occurred. [5][8] have also confirmed the advantages of adaptive approaches.

Our algorithm is shown in Fig. 2. At the fault diagnosis phase, we assume that, the probing results from fault detection are already known.

The function getSelected(T) returns all the selected probes since last time the information gain of T was updated. The function isDependent(Queue, T) returns whether or not T is dependent on any given probe in Queue. The function updateGain(T, **D**) calculates the information gain of T given **D**, and demines the running time of the algorithm.

Initially, all the candidate probes are evaluated, and **S** is sorted in descending order of the information gain. T_{max}

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

represents the probe whose information gain is currently the largest. MaxGain stores the current largest information gain, and the corresponding probe is arg(MaxGain). For simplicity, we use T_k instead of arg(MaxGain) here.

First T_k is set to be the probe with the largest information gain in **S** (i.e. second largest in the last round). If no probe dependent with it has been selected since its last update, T_k need not to be reevaluated. Further, it can be selected directly without reevaluation of other candidates. Otherwise, update its information gain. If the updated value is greater than that of the next probe in **S**, for instance, T_l, again T_k can be selected directly without reevaluation of other candidates. Otherwise, we have to evaluate whether or not T_l's information gain needs to be updated, and then compare T_l's with T_k's gain. If T_k's gain is greater, T_k is selected; otherwise T_k is set to T_l and MaxGain is updated. This process is repeated until the information gain of T_k is greater than that of the next probe in **S**.

The implementation of the two key functions is as follows.

(1) Implementation of isDependent():

We use the following theorem to simplify the evaluation of whether two variables in a Bayesian network are independent.

Theorem 4 ([17]). Let **G** be a directed acyclic graph, and $\mathbf{G^m}$ is the moral graph of **G**. let **X**, **Y** and **Z** be three different sets of variables where each intersection of any two sets is empty. **Z** d-separates **X** and **Y** in **G** if and only if **Z** u-separates (undirected separates) **X** and **Y** in $\mathbf{G^m}$.

Here, u-separate means if **Z** is removed from $\mathbf{G^m}$, then there is no path between **X** and **Y** in $\mathbf{G^m}$. We use this theorem to evaluate conditional independence in Bayesian networks. Firstly, the moralized counterpart $\mathbf{G^m_{bayes}}$ of a Bayesian network $\mathbf{G_{bayes}}$ (directed acyclic graph) is formed by connecting nodes that have a common child, and then makes all edges undirected. Secondly, the nearly determined nodes and their related edges are removed. Thirdly, if there is no path between two probe nodes in the left graph, these two probes are independent. In fact, after removing some nodes and their connected edges, usually $\mathbf{G^m_{bayes}}$ turns into a certain number of connected graphs. The probes in the same connected graph are dependent, otherwise they are independent.

(2) Implementation of updateGain():

We implement updateGain() on the basis of [12] in which the authors proved that:

$$G(T|\mathbf{D}) = -\sum_{T,\mathbf{X_{paT}}} P(\mathbf{X_{paT}}, T|D)\log P(T|\mathbf{X_{paT}}) + \sum_{T} P(T|\mathbf{D})\log P(T|\mathbf{D}) \quad (6)$$

where $\mathbf{X_{paT}}$ refers to T's parent nodes, i.e. the network nodes that T depends on. Based on this equation, we can calculate G(T|**D**) directly instead of determining H(**X**|**D**) and H(**X**|**D**,T), and their difference respectively. Especially, because the number of states of T's parents is much less than that of the system, the speed of the algorithm will be greatly improved. Existing algorithms for Bayesian networks can be used to compute $P(\mathbf{X_{paT}}, T|\boldsymbol{D})$ and P(T|**D**).

V. Evaluation

We have implemented BPEA [12], the best solution as far as we know, and our algorithm IAP. We make comparisons of these two algorithms on realistic Internet-like topologies under multiple-fault scenarios. The experiment setup is similar to that of [12].

A. Implementation of Algorithms

Zheng et al. [12] proposed an approximation approach which utilizes the belief propagation (BP)[13]. The formulation in (6) modifies BP by replacing the marginal posterior P(**X**| **D**) with its factorized BP approximation, and makes use of the BP message passing mechanism to perform summation over **X**. Because BP message updates are done locally, the posterior entropy for all unobserved probes can be computed during a single application of BP. Hence, selecting the next probe requires only one run of the BPEA approximation algorithm, which makes it is much faster than regular greedy algorithms. We implement BPEA on top of the belief propagation in Kevin Murphy's Bayes Net Toolbox [18].

We make use of the junction tree [18] to update information gain of candidate probes in our algorithm.

B. Experiment setup

Generating Networks: The network topology influences the symptoms observed. Odintsova et al. [19] discussed topology effects on fault diagnosis in computer networks. There are two well-known types of network topologies: classical random graphs [20] and power-law [21] networks. Random graphs are considered as a baseline model, while power-law models received significance in recent years and have been applied in various contexts including the Internet, the WWW, P2P systems and Social Networks among others [19].

Based on the above considerations, we compare the two algorithms under these two types of topology. The obtained results are similar. Therefore and due to space limitation, we only present the results in power-law networks. We use INET [22] to create power-law networks. A node represents a physical device, or a logical entity such as a software application.

Generating Candidate Probes: Given a topology, we can randomly select a number of nodes as probe stations. We set the number of probe stations to be 3 in our experiments. If more or less probe stations are deployed, the number of candidate probes and computation time will increase or decrease correspondingly. In order to reflect common practice in current networks, we assume that the routing between probe stations and network nodes follows the shortest path.

From a complete set of candidate probes, a smallest subset of probes is selected for fault detection using the greedy

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

search approach [3] that ensures every node is covered by at least one probe. The remaining probes are considered candidates for fault diagnosis. After removing redundant ones, for example, in a network with 500 nodes there are on average 524 candidate probes left for fault diagnosis.

Injecting Faults and Noisy observations: To inject faults, we allow each node to break down independently with its prior probability. Murphy et al.[14] point out that the prior probability of a diseased node being on is extremely low in real networks (typically of the order of 10^{-3}). Because the exact value of prior probability is usually unkown, for each node we randomly generate a prior probability with aNormal distribution $N(0.005, 0.003)$. When making inferences, we simply assume every node has the same prior probability 0.005.

Similar to [12], we assume a probe is a noisy-OR test. The joint probability distribution of all the tests and network nodes forms the well-known QMR-DT model [23]:

$$P(T_j = 0 | X_1, X_2, \dots, X_k) = (1 - \rho_{i0}) \prod_{x_i} \rho_{ij}^{x_i}$$

Here, $X_1, X_2 \dots X_k$ are the network nodes on T_j's path. Every probe T_j and node X_i are associated with a noise parameter ρ_{ij}, also called inhibition probability. It is usually a small probability that a probe succeeds even when a node on its path has failed. The inhibition probability is a measurement of the amount of noise in the network. There is another parameter ρ_{i0} called the leak probability which accounts for the cases of a probe failing even if all the nodes on its path are OK.

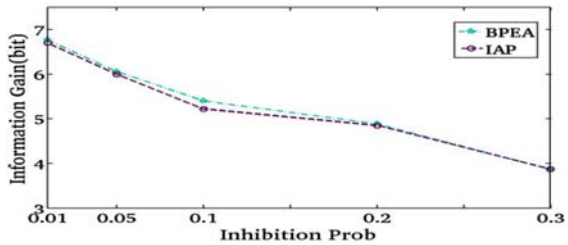
For the sake of simplicity, we assume the cost of a single probe is the same and additive, thus we can simply limit the total costs to be no more than k probes. Here we set k to be 30. All the figures show results averaged over 100 runs.

C. Diagnostic quality

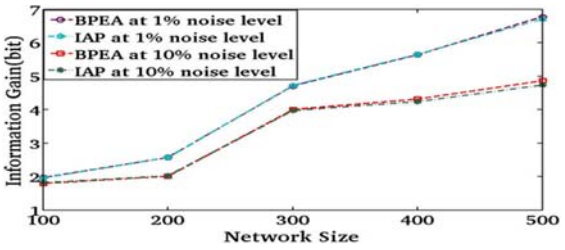
Fig. 3(a) compares the total information gain by BPEA and IAP at various noise levels. The network has 500 nodes. When the noise level is low, nearly all of the probes can accurately reflect the ground truth, thus acquire relatively more information. As noise level goes up, the information gain goes down because spurious responses actually increase uncertainty in the system. From another point of view, Fig. 3(a) shows that more probes are needed to ensure accuracy of diagnosis as noise level increases.

The BPEA's information gain is a little higher than that of IAP. But the average difference does not exceed 1% even when the noise level reaches 30%. The authors of BPEA have demonstrated that their algorithm's average approximation error do not exceed 2%, which means that the one of IAP does not exceed 3% compared to the exact value. The main reason of approximation error of IAP is that although some probes are considered approximately conditionally independent, in fact they have weak impact on each other. The omitted information exchange results in errors.

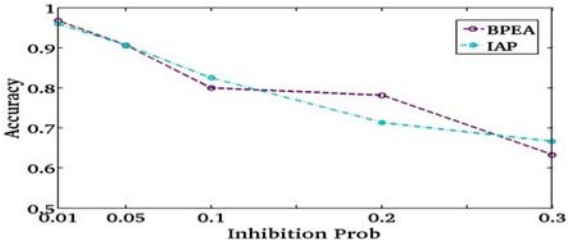
Fig. 3(b) shows the information gain of BPEA and IAP at 1% and 10% noise levels for different network sizes. As network size increases, the same number of probes can acquire more information. From another perspective, it says that more probes are needed to ensure accuracy of diagnosis in larger networks.



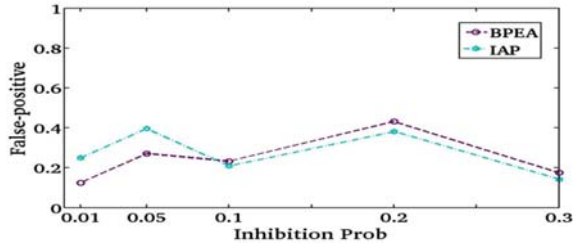
(a) Total information gain at various noise levels in a network with 500 nodes



(b) Total information gain in networks with different sizes



(c) Diagnosis Accuracy by taking responses of selected probes as input



(d) Diagnosis Precision by taking responses of selected probes as input

Fig. 3. **Diagnostic quality:** Comparisons of BPEA and IAP under various network size and noise level.

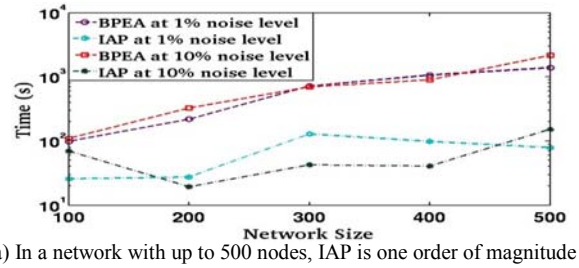
After selection is completed, we take the probing responses (including initial results from fault detection) as input of the same fault diagnosis algorithm. Intuitively more informative observations should lead to better diagnosis.

We use two metrics, namely accuracy and false positive to evaluate diagnosis results. Let $\boldsymbol{F}$ be the set of real faults, $\bar{\boldsymbol{F}}$ the set of good nodes, and $\boldsymbol{H}$ the set diagnosed to have failed:

$$\text{Accuracy} = |\boldsymbol{H} \cap \boldsymbol{F}|/|\boldsymbol{F}| \qquad \text{False positve} = |\boldsymbol{H} \cap \bar{\boldsymbol{F}}|/|\boldsymbol{F}|$$

For objectivity reasons, we select a diagnosis algorithm from a third party, namely Shrink[15]. Although we have tried other alternatives such as BayesNet[24], Maxcoverage[25] and some algorithms in the Bayes Net Toolbox [18], Shrink's

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.



(a) In a network with up to 500 nodes, IAP is one order of magnitude faster than BPEA and more tolerant to noises.

Number of Probes evaluated in each iteration	Number of Nodes / Candidate Probes				
	100/ 108	200/ 205	300/ 309	400/ 410	500/ 524
1% noise level	1.3	0.4	0.4	0.4	0.4
10% noise level	1	0.4	0.3	0.5	0.5

(b) We give the average number of candidate probes as a reference. For example, regular greedy methods and BPEA need evaluate 524 candidates during each iteration in a network with 500 nodes, but on average, ours need to update the information gain of only 0.4 probe.

Fig. 4. **Speed**: Running time and the number of evaluations saved

performance is the best in large scale and noisy networks.

Fig. 3(c) compares diagnosis accuracy at various noise levels in a network with 500 nodes. In generally, the diagnosis accuracy on the basis of BPEA is roughly equivalent to that on the basis of IAP. As noise level goes up, the diagnosis quality declines accordingly. Because we limit the total number of probes to be 30, the accuracy is not very high. The accuracy increases if we send out more probes. Fig. 3(d) compares diagnosis precision in the same settings. Similar to accuracy, the precision on the basis of either input is quite close to one another. From Fig. 3(c) and (d), it can also be seen that the reduction of entropy precisely measures the amount of information that observations bring for fault diagnosis.

D. Speed

From Fig. 4(a) we can see that compared to BPEA, the running time of IAP is more tolerant to noise. But basically the noise level doesn't have great impact on the speed. BPEA's running time increases rapidly with the network size. That's because if the size of the network is big, the propagation of BP messages takes a long time to converge. There is nearly no difference in the speed of these two algorithms in a network with 100 nodes. But IAP is one order of magnitude faster than BPEA when network size goes up to 500 nodes. BPEA takes approximately 37 minutes to select the 30 most-informative probes, which is not applicable for real-time fault diagnosis in large scale networks. In comparison, IAP needs only 3-4 minutes in the same settings.

The running time saved by IAP lies in the fact that the number of probes required to be evaluated in every iteration is significantly reduced, as shown in Fig. 4(b). We have also implemented a regular greedy algorithm but which proven to be slow. Approximately, even BPEA is two orders of magnitude faster than the regular greedy one.

IV. Related Work

Tang et al. [6][7] proposed a probabilistic fault diagnosis technique called AIR that integrates active monitoring with passive fault reasoning. The AIR first passively analyzes observed alarms and generates fault hypotheses. If the correlated symptoms are sufficient to explain the fault hypotheses (i.e. high fidelity), the fault reasoning process terminates. Otherwise, a set of tests will be performed to acquire the most likely unobserved symptoms that can contribute to the fault hypothesis fidelity. The authors model this problem as a weighted set-covering problem and developed a heuristic greedy set-covering approximation algorithm. The function to differentiate probes is as follows: $R_i = |S_{a_i}| / \sum_{s_j \in S_{a_i}} w_{ij}$, in that $|S_{a_i}|$ is the number of symptoms the probe triggers and w_{ij} is the cost. After receiving probing responses, the fidelity of fault hypotheses is reevaluated.

The selection of optimal tests has been addressed in various areas including diagnosis, monitoring, decision analysis, etc. For example, Song et al. [26] apply Bayesian experimental design to select active measurements that maximize the amount of information about network path properties, and apply inference techniques to solve end-to-end performance monitoring problems. Their approach applies to any additive performance metrics, such as delay or loss rate.

Natu et al. [8][9] present an adaptive probing solution for fault diagnosis. Based on results of fault detection, a set of suspected nodes that consist of all nodes on the failed probes paths is built. For each suspected nodes, a function is employed to select the best informative probe: $T^* = argmax\left\{P(T|s) + \left(1 - \left(\cup_{n \in \{ShadowNodes - s\}}(\mathrm{T}|\mathrm{n})\right)\right)\right\}$. In that s represents the suspected node, and $1 - P(\cup_{n \in \{ShadowNodes - s\}}(\mathrm{T}|\mathrm{n}))$ represents the probability that probe T will only pass s but not other suspects. Therefore, the probe that will traverse the least number of suspected nodes will be picked out as the next probe. This method is vulnerable to noises, because if a failure response gets lost, all the nodes it depends will not be treated as suspected nodes. Further, the selection of T* doesn't consider relationships between probes.

VII. Conclusion

In this paper, we extend our fault diagnosis by studying how to select an optimal set of probes. Exploiting structural property of Bayesian networks, we proposed a theorem and two approaches based on which a highly efficient algorithm is presented. Experiments demonstrate indeed that our algorithm is 3 orders of magnitude faster than the regular greedy method and 10 times faster than the best known solution at the cost of negligible loss in diagnosis quality.

Monitoring is fundamental to nearly all functional areas of network and service management. We believe that

contributions presented in this paper have a significant potential and can be applied in many of the network and service management areas.

ACKNOWLEDGEMENT

This work was supported in part by the 973 project of China (2007CB310703), Funds for Creative Research Groups of China (60821001), NSFC (60973108) and 863 Project (2008AA01Z201).

REFERENCES

[1] G. Jakobson and M. Weissman. Alarm Correlation. *IEEE Network*, 1993.
[2] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie. A Coding Approach to Event Correlation. In *Proc. IM'95*, Pages 266-277, 1995.
[3] M. Brodie, I. Rish, S. Ma, N. Odintsova, and A. Beygelzimer. Active Probing Strategies for Problem Diagnosis in Distributed Systems. In *Proc. IJCAI'03*, 2003
[4] I. Rish, M. Brodie, N. Odintsova, S. Ma, and G. Grabainik. Real-time problem determination in distributed systems using active probing. In *Proc. NOMS'04*, pages 133–146, 2004.
[5] I. Rish, M. Brodie, S. Ma, N. Odintsova, A. Beygelzimer, G. Grabainik, and K. Hernandez. Adaptive Diagnosis in Distributed Systems. *IEEE Trans. Neural Networks*, 16(5):1088-1109, 2005.
[6] Y.Tang, E. Al-Shaer, and R. Boutaba. Active Integrated Fault Localization in Communication Networks. In *Proc. IM'05*, France, 2005.
[7] Y. Tang, E. Al-Shaer, and R. Boutaba. Efficient fault diagnosis using incremental alarm correlation and active investigation for internet and overlay networks. *IEEE Trans. Network and Service Management*, 5(1): 36-49, 2008
[8] M. Natu and A. S. Sethi. Probabilistic Fault Diagnosis Using Adaptive Probing. In *Proc. DSOM*, LNCS 4785, pp. 38–49, 2007.
[9] M. Natu and A. S. Sethi. Application of Adaptive Probing for Fault Diagnosis in Computer Networks. In *NOMS'08*, pp:1055-1060, 2008.
[10] L. Cheng, X.S. Qiu, L.M. Meng, Y. Qiao and Z.Q. Li. Probabilistic Fault diagnosis for IT Services in Noisy and Dynamic Environments. In *Proc.* IM'09, New York, June 2009.
[11] D. V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27:986–1005, 1956.
[12] A. X. Zheng, I. Rish, and A. Beygelzimer. Efficient test selection in active diagnosis via entropy approximation. In *Proceedings of Uncertainty in AI (UAI' 05)*, 2005
[13] J. Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Mateo, California, 1988.
[14] K. P. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of Uncertainty in AI(UAI' 99)*, pages 467-475, 1999.
[15] S. Kandula, D. Katabi, and J. P. Vasseur. Shrink: A tool for failure diagnosis in IP networks. In *Proc. ACM SIGCOMM MineNet Workshop*, Aug. 2005.
[16] S. Kullback. The Kullback-Leibler distance. *The American Statistician*, 41:340–341, 1987.
[17] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, D. J. Spiegelhalter. Probabilistic Networks and Expert Systems. Springer-Verlag, New York, 1999
[18] K. P. Murphy. The Bayes Net Toolbox for Matlab. In *Computing Science and Statistics: Proceedings of the Interface*, volume 33, 2001.
[19] N. Odintsova and I. Rish. Empirical Study of Topology Effects on Diagnosis in Computer Networks. In *Proc. Mobile Adhoc and Sensor Systems*, 2007.
[20] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617~1622, 1988.
[21] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. *ACM SIGCOMM Computer Communication Review*, 29(4):251~262, 1999.
[22] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report CSE-TR-456-02, University of Michigan, 2002.
[23] T. Jaakkola and M. Jordan. Variational probabilistic inference and the qmr-dt database. *Journal of Artificial Intelligence Research*, pages 291–322, 1999.
[24] M. Steinder and A.S. Sethi. Increasing Robustness of Fault Localization through Analysis of Lost, Spurious and Positive Symptoms. In *Proc. IEEE INFOCOM*, 2002.
[25] R. R. Kompella, J. Yates, A. Greenberg, and A. Snoeren. Detection and Localization of Network Black Holes. in *Proc. IEEE INFOCOM*, 2007.
[26] H.H. Song, L.L. Qiu and Y. Zhang. NetQuest: A Flexible Framework for Large-Scale Network Measurement, in *Proc. ACM SIGMETRICS*, Saint-Malo, France, June 2006.

APPENDIX

Proof of Theorem 3.

The inequality to be proven is equivalent to:

$$H(\boldsymbol{X}|\boldsymbol{D}^{(i)}) - H(\boldsymbol{X}|\boldsymbol{D}^{(i)}, T_j) \le H(\boldsymbol{X}|\boldsymbol{D}^{(i-1)}) - H(\boldsymbol{X}|\boldsymbol{D}^{(i-1)}, T_j)$$

The left-hand side is:

$$\begin{aligned}
&-\sum_{\boldsymbol{X},\boldsymbol{D}^{(i)}} P(\boldsymbol{X},\boldsymbol{D}^{(i)}) log P(\boldsymbol{X}|\boldsymbol{D}^{(i)}) \\
&+\sum_{\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}} P(\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}) log P(X|\boldsymbol{D}^{(i)},T_j) \\
&=-\sum_{\boldsymbol{X},\boldsymbol{D}^{(i)},\mathrm{T_j}} P(\boldsymbol{X},\boldsymbol{D}^{(i)},\boldsymbol{T_j}) log P(\boldsymbol{X}|\boldsymbol{D}^{(i)}) \\
&+\sum_{\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}} P(\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}) log P(\boldsymbol{X}|\boldsymbol{D}^{(i)},T_j) \\
&=-\sum_{\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}} P(\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}) log \frac{P(\boldsymbol{X}|\boldsymbol{D}^{(i)})}{P(\boldsymbol{X}|\boldsymbol{D}^{(i)},T_j)}
\end{aligned}$$

Similarly, the right-hand side is:

$$\begin{aligned}
&-\sum_{\boldsymbol{X},T_j,\boldsymbol{D}^{(i-1)}} P(\boldsymbol{X},T_j,\boldsymbol{D}^{(i-1)}) log \frac{P(\boldsymbol{X}|\boldsymbol{D}^{(i-1)})}{P(\boldsymbol{X}|\boldsymbol{D}^{(i-1)},T_j)} \\
&=-\sum_{\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}} P(\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}) log \frac{P(\boldsymbol{X}|\boldsymbol{D}^{(i-1)})}{P(\boldsymbol{X}|\boldsymbol{D}^{(i-1)},T_j)}
\end{aligned}$$

When subtracts the left from the right, after some regular calculations, we have:

$$\begin{aligned}
&-\sum_{\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}} P(\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}) log \frac{P(\boldsymbol{X}|\boldsymbol{D}^{(i-1)})P(\boldsymbol{X}|\boldsymbol{D}^{(i)},T_j)}{P(\boldsymbol{X}|\boldsymbol{D}^{(i-1)},T_j)P(\boldsymbol{X}|\boldsymbol{D}^{(i)})} \\
&=\sum_{\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}} P(\boldsymbol{X},T_j,\boldsymbol{D}^{(i)}) log \frac{P(T_j,\boldsymbol{D}^{(i)})}{P(\boldsymbol{D}^{(i)})P(T_j|\boldsymbol{D}^{(i-1)})} \\
&=\sum_{T_j,\boldsymbol{D}^{(i)}} P(T_j,\boldsymbol{D}^{(i)}) log \frac{P(T_j,\boldsymbol{D}^{(i)})}{P(\boldsymbol{D}^{(i)})P(T_j|\boldsymbol{D}^{(i-1)})}
\end{aligned}$$

By Kullback-Leibler distance [16],

$$\sum_{T_j,\boldsymbol{D}^{(i)}} P(T_j,\boldsymbol{D}^{(i)}) log \frac{P(T_j,\boldsymbol{D}^{(i)})}{P(\boldsymbol{D}^{(i)})P(T_j|\boldsymbol{D}^{(i-1)})} \ge 0$$

With equality if, and only if, the numerator equals the denominator of log(·), i.e.: $P(T_j,\boldsymbol{D}^{(i)}) = P(\boldsymbol{D}^{(i)})P(T_j|\boldsymbol{D}^{(i-1)})$

Because: $P(T_j,\boldsymbol{D}^{(i)}) = P(\boldsymbol{D}^{(i)})P(T_j|\boldsymbol{D}^{(i-1)},\mathrm{T_i})$, Hence, the equality holds if and only if given $\mathbf{D}^{(i-1)}$, $\mathrm{T_i}$ and $\mathrm{T_j}$ are independent, i.e.: $P(T_j|\boldsymbol{D}^{(i-1)}) = P(T_j|\boldsymbol{D}^{(i-1)},\mathrm{T_i})$. ■