Directional Diagnosis for Wireless Sensor Networks

Wei Gong[1,2], Kebin Liu[2,3], Yunhao Liu[2,3], Xibin Zhao[2], Ming Gu[2]
[1]Department of Computer Science and Technology, Tsinghua University
[2]Key Laboratory of Information System Security of Ministry of Education, TNLIST, School of Software, Tsinghua University
[3]Department of Computer Science, Hong Kong University of Science and Technology

***Abstract*—Network diagnosis is crucial in managing a wireless sensor network (WSN) since many network-related faults, such as node and link failures, can easily happen. Diagnosis tools usually consist of two key components, information collection and root-cause deduction, while in most cases information collection process is independent with root-cause deduction. This results in either redundant information which might pose high communication burden on WSNs, or incomplete information for root-cause inference that leads false judgments. To address the issue, we propose DID, a directional diagnosis approach, in which the diagnosis information acquirement is guided by the fault inference process. Through several rounds of incremental information probing and fault reasoning, root causes of the network abnormalities with high credibility are deduced. We employ a node tracing scheme to reconstruct the topical topology of faulty regions and build the inference model accordingly. We implement the DID approach in our forest monitoring sensor network system, GreenOrbs. Experimental results validate the scalability and effectiveness of this design.**

***Keywords*—fault detection; directional diagnosis; wireless sensor networks**

I. INTRODUCTION

An increasing large number of Wireless Sensor Networks (WSNs) have been applied in many different applications such as habitat surveillance, infrastructure protection and clinical monitoring [1][2][3][3]. Typically, sensor nodes are usually provisioned with low-capacity batteries and work in an ad-hoc and energy conservative manner. Although the connectivity and functionalities of WSNs have been greatly improved by many efforts [5][6][7][24][25], sensor networks still suffer many network-related faults, e.g., failure nodes or lossy links [15][16] due to the error-prone nodes and other uncertainties, like environmental interference. The faults in deployed WSNs are usually difficult to detect and localize due to its improvisational nature and invisibility of internal running status. Therefore, the design of effective online WSN diagnosis tools, which can help network administrators monitor the network operational status and maintain a sensor network system, has drawn significant attentions.

Existing works on fault diagnosis in WSNs typically consist of two independent parts: collecting running information from nodes and deducing root causes of network exceptions. For instance, Sympathy [16] periodically collects run-time status from all sensor nodes and leverages a decision-tree based scheme to find the most possible causes of observed network exceptions. Khan et al. [8] propose to use external power measurements to accumulate all senors' power information. A classifier is then applied to determine the internal health condition of an unresponsive host and the possible cause of its failure. Such methods, however, share a common drawback: their diagnosis information gathering process is static and preplanned without the aid of useful intermediate reports. Clearly, the pre-determined scheme may either result in redundant information, posing unnecessary communication burden, or incompleteness of information for root-cause inference so that it leads to a large number of false results.

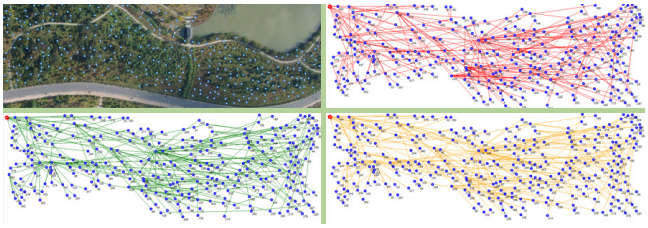


Fig.1. GreenOrbs project deployment and dynamic topologies

To address the above issues, we propose an online diagnosis approach, DID, in which information acquirement is directed by the intermediate results of a probabilistic inference model. Combining an incremental probing scheme and a dynamic probabilistic inference model, DID effectively localizes the root causes of various network abnormalities. Differing prior methods, this approach dynamically recognizes the most useful information for further diagnosis according to current results of the fault inference engine. Additionally, the diagnosis is confined to topical area covering the problematic network elements in high potential. With a low overhead, a high-accuracy real-time network diagnosis service is thus provided.

Indeed, probing is a widely used diagnosis technique for obtaining internal network status in the Internet and enterprise networks. Several types of probing schemes have been proposed for different applications, while most of them rely on expert knowledge or prior information like network topology. Besides, the network models are often assumed remaining unchanged during the running phase [17]. Compared to the Internet and enterprise networks, however, sensor networks have the following unique features: 1) Sensor nodes usually have limited computational resources and energy supply; 2) Due to the environment interference and uncertainty of the wireless medium, the network topology is highly dynamic. What is more, prior information is difficult to be acquired since WSNs are self-organized. As show in Fig.1, the up-left corner

is the physical deployment of our GreenOrbs project; the other three topology figures are snapshots at different time that shows that link statuses between nodes change from time to time; 3) As the sensor nodes are error-prone, the number of simultaneously failures is hard to predict in advance, while in enterprise networks it is often assumed that this number is no more than a predefined constant k. Thus, existing approaches cannot be directly applied to WSNs because no information about inner dependencies among network elements is really available, and frequently changed topologies further make the schemes infeasible for WSNs.

The major contributions of this work are as follows.

1) According to the feature of dynamic topology in WSNs, we propose a node tracing scheme to recover topical topology which only includes nodes involved in the faulty region.

2) We introduce an incremental probing scheme in which the selection of next probe is based on the results of current fault inference.

3) We implement our diagnosis approach, DID, and verify its effectiveness in our forest monitoring project, GreenOrbs. The results of our field test show that the root causes are effectively and accurately localized.

4) We conduct extensive simulations under varied conditions to evaluate the scalability and effectiveness of DID design.

The rest of this paper is organized as follows. Section II introduces related works. Section III describes the architecture of our design. The node tracing scheme is detailed in Section IV. We discuss the incremental probing scheme together with the inference model in Section V. In Section VI, we present our simulation and implementation results. We conclude this work in Section VII.

II. Related Work

Most of existing approaches for sensor network diagnosis follow similar patterns: they gather information from sensor nodes in predetermined manner and then use various inference algorithms which take collected information as inputs to deduce root-causes of network exceptions. Some researchers propose to scan the residual energy and other metrics of each sensor node in the running time [9], so that the sink can analyze the gathered information to obtain network operational conditions. Sympathy [16] periodically collects pre-defined in-network information from each sensor node such as traffic flow and neighbor list and uses an empirical decision-tree based scheme to root-cause the network failures. It selects an optimal set of information metrics to minimize the communication cost. A passive diagnosis approach, PAD, which passively observes the network symptoms from the network, is proposed for sensor network by Liu et al. [18]. Using the marks collected at the sink, a probabilistic inference model can be dynamically constructed and maintained which takes both positive and negative symptoms as inputs and reports the inferred posterior probability of possible root-causes. Khan et al. [8] propose to use an independent power-metering subsystem which can collect power consumption traces of high-end sensor nodes. Most of exiting works, if not all, however, fail to link the fault inference process and the diagnosis information gathering process.

Detecting and localizing network anomalies is an important issue in Internet and enterprise networks, and many efforts have been devoted to this field. Bejerano and Ragosti propose to actively monitor all the links of a network in an efficient way [10]. They form the problem of selecting a minimal set of paths which covers all links in network as an instance of the well-known NP-hard problem, minimal set cover, and employ a heuristic algorithm to solve it. Duffield proposes a tomographic method in [11], which can infer the location of problematic links assuming binary network performance characteristics. Dhamdhere et al. [12] extend the work of Duffield by using additional network measurements. Score [13] introduces a shared risk model in which a two-level graph is simplified for inference model and also formulates the problem of localizing fault as a NP-hard problem. By exploring the bipartite graph inference model, Kandula et al. [14] reduces the complexity of the inference process. In [20], Sherlock proposes a multistate and multilevel inference graph for the network diagnosis and a scoring function to determine root causes for observed exceptions. The selection of optimal probes has been addressed in [19], where Song et al. apply Bayesian experimental design to select active probes for maximizing the amount of information about path properties. The above schemes either require prior information of the network dependencies such as topology or take simplified methods to model the network dependencies. However, highly dynamic and self-organizing properties of WSN make it impractical to assume that network topology can be acquired in advance and remain relatively unchanged during running time. Also, the assumption that there are no more than a small constant number of failures in Internet and enterprise networks which can avoid NP-hard computation complexity is also invalid for WSNs.

III. System Architecture

We consider sensor networks where sensing data are periodically sent from source nodes to a sink through multi-hop communication. There could be both full-function nodes which involve in both communication and reporting sensing data and single-function nodes which only relay packets for others in the sensor networks.

We design a directional diagnosis approach, DID, for such sensor networks. The goal of DID is to aid network administrator in discovering the root causes of the network abnormal phenomenon. DID implants a tiny tracing sniffer into each node in WSNs which records packets relay information so that sink can collect these information from nodes to reassemble topical topology for further diagnosis when there are network exceptions happened. Usually the network exceptions are predefined and used as a symbol of fault detected which triggers diagnosis process. The definition of exception can be in various criteria according to specific application requirement, such as a long time delay of data arrival or insufficient amount of data. However, only tracing information from sniffers does not suffice to precisely infer and localize the root cause of network faults. Thus by employing an

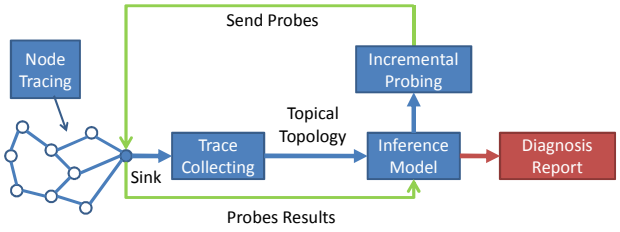


Fig.2. DID architecture overview

incremental probing scheme, we continuously refine the inference model to derive network element statuses more exactly.

As illustrated in Fig.2, DID is mainly composed of four components: a node tracing module, a trace collecting module, a probabilistic inference model and an incremental probing module. The node tracing module resides in each node and counts the source of all packets passing by. At the sink, the tracing collection module gathers above tracings from nodes which participate in communication with suspiciously faulty nodes when any network exception is discovered. Then the coarse inference model can be built on the topical topology reconstructed. The inference model guides the selection of next probe sent to sensor networks. At the same time with the help of probing results, the inference model is continuously updated. When incremental probing stops, the diagnosis report which contains the posterior probability of network elements being faulty is yielded.

IV. NODE TRACING

The network structure of WSN frequently changes due to the innate self-organization; therefore prior information about topology is hard to be obtained for constructing the inference model which can be easily built in Internet or enterprise networks. The topology of WSN is highly dynamic during the runtime; so the network statuses also need to be acquired continually to reflect network changes in real time. To address the above requirements, we propose a node tracing algorithm which dynamically reconstructs topical topology and derive the inner dependencies among network elements.

A. *Tracing Scheme on Sensor Nodes*

Every node in networks maintains a relay list and a trace list for its downstream source nodes. Each entry of relay list is composed of a previous-hop neighbor ID and the number of packets relayed for that neighbor. And each entry of trace list consists of a source node ID and the newest sequence number of the received packet from that source.

As shown in Algorithm 1, upon receiving a packet, the intermediate node first checks relay list. If there is no entry for the previous hop node ID, it creates a new entry for it and sets the number of relayed packet to 1. Otherwise, it just increase the packets relay number by 1 in the corresponding entry. Then it checks the trace list. A new entry for this source node is created in its cache and the sequence number for the packet is recorded if there is no entry for source ID of the packet. If there exists an entry in the trace list for the source node and the sequence number in the packet is larger than trace list entry, the intermediate node updates the trace list entry with the new sequence number. The sink also participates in the tracing process. However, the only difference is that the sequence number of entry in the trace list is the oldest at the sink while other nodes record the newest. When one-time topical diagnosis completed, we should reset sequence number of the faulty nodes diagnosed which indicates a new start time point.

Algorithm 1 Node_Tracing (Packet p)

```
1  check relay list for previous hop node ID of packet P;
2  if no entry for packet p
3      create entry with previous hop node ID and set
   number of replayed packet to 1;
4  else correspondently increase the number of relayed
   packet by 1;
5  end if
6  check trace list for source node ID of packet P;
7  if no entry for packet p
8      create entry with source node ID and sequence
   number of p;
9  else if sequence number in the list is large than p's
10         return;
11     else update sequence number in correspondent
   entry;
13     endif
14 endif
15 return;
```

B. *Collecting Trace from Sensor Nodes*

At the sink, trace collecting process is trigged by some network exceptions happened. One of the widely used criteria is that insufficient amount of sensing data from nodes during some time interval indicates possibly some network faults detected. Then the sink extracts the rows of these suspicious nodes from the trace list and then packages this information as a collecting probe which is broadcasted to networks. This flooding scheme is different from normal broadcasting because only if the trace list of nodes contains any suspicious node, the probe is rebroadcasted to neighbor nodes. Otherwise the probing is stopped on that node. By using this controlled flooding, additional traffic overhead incurred by probe broadcasting is effectively restricted.

Algorithm 2 ReplyProbe (Probe p)

```
1  for each row in p
2    if row.ID is in the trace list
3        if row.sequence_number is larger than its
   correspondent in trace list
4            send thisNode.relay_list and neighbor list to
   sink;
5            broadcast p to its neighbor;
6            break;
7        end if
8    endif
9  return;
```

As shown in Algorithm 2, on receiving a probe, each node needs to check whether it has ever relayed packet for any suspicious node. It firstly examines whether there is suspicious

node ID in its trace list or not. If so, it then compares the sequence numbers of row in the probe and trace list. If the sequence number in the probe is small, then the node should hand in the trace information which contains relay list and one-hop neighbor list which is often easily obtained from down-level network protocols such as routing protocol, and then broadcast this probe to neighbor nodes. Otherwise, larger sequence number in probe *p* means that this node had relayed packets for suspicious nodes before the problem time interval in which network exceptions happened, so it is not involved in the diagnosis process. The collected relay list is used to facilitate the construction of the inference model as it tells the strength of the dependency between the parent and its successive nodes.

After gathering demanded information, the sink starts to reconstruct the topical topology for diagnosing. The neighbor lists and suspicious nodes set are used to recover topology. The reconstruction starts from the sink, then we connect all neighbor links between each other in the replied neighbor lists. Obviously, there might be some irrelevant edges and nodes included. Therefore, we need to remove those irrelevant network elements. Thus we iteratively cut off those nodes whose degree is one and their connected edges until all unqualified leaf nodes are removed.

V. INCREMENTAL PROBING

We build a probabilistic inference model which encodes the dependencies among different network elements, such as node status, link status and sensing function, based on the reconstructed topology from node tracing module. Exterior symptoms like loss of data samples or link status are considered as inputs. When specific symptoms are observed by our probing algorithm, we can reason the posterior probability of the failures of each network element and find the most probable explanation for observed symptoms.

Most existing diagnosis schemes for sensor networks have no feedbacks on diagnosis information collecting. In our approach, inference engine starts when network abnormality is detected. However, as the initial observed symptoms is not sufficient to root-cause the real faults, we use incremental probing scheme to discover the most critical symptoms that are important to explain the problem. The selection of probe which is used to collect more information about network elements is under the guide of the newest updated inference model.

A. Inference Model

As no prior information about topology can be acquired in advance and topologies are also highly dynamic in running time, most existing probabilistic inference schemes for Internet and static enterprise networks which use the bipartite graph or tree-based inference models are not applicable in WSNs. Here, we employ a multi-level inference model which is similar to [18] to encode the inner dependencies in sensor networks.

A Belief Network is a directed acyclic graph (DAG) that consists of a number of vertices which denote random variables and edges which denote causal relations between variables. A directional arc from vertex ***A*** to vertex ***B*** means ***B*** is the outcome of cause ***A***. Each vertex in belief networks is assigned with its probability distribution. The difference is that the vertex which has no parent is assigned prior probability, we call it *CAUSE* for short, and the other intermediate vertex is assigned conditional probability distribution (CPD), we call it *SYMPTON*. Given certain evidence which might contain several *SYMPTONs*, the Belief Network can deduce posterior probability of all *CAUSE*s.

1) Belief Network construction

In our approach, the Belief Network structure is constructed from the topical network topology obtained from the node tracing module. An example topology is depicted in Fig. 3(a) which is composed of a sink and two sensor nodes. The directional edge between two nodes denotes a wireless data transmitting link. There are six types of vertices in our Belief Network, each of which has two statuses: UP and DOWN. We explain each type as follows. The radio vertex indicates the radio communication function of the correspondent sensor node. It impacts related link status. The sensing vertex indicates the sensing function of the node. The connection vertex describes network connectivity from sensor node to the sink. The path vertex stands for network condition of specific course. The link vertex represents the communication conditions between two nodes. The data report vertex denotes the status of the data reception of the source node at the sink. For example, if the sink observes insufficient sensing data in a period of time or a long time delay from a source, this vertex is observed as DOWN status which may trigger a diagnosis process. Note that, the relay node which has no sensing function should not contain this vertex. The status of the data report vertex depends on two parents, the sensing vertex and the connection vertex; while the link vertex status is decided by the status of two neighbor nodes. It is obvious that the dependency among connection, path and link vertex is deduced from communication topology. According to above rules, we can obtain a multi-level belief network is composed of these six types of variables, as shown in Fig.5 (b). Among the six types of variables, the statuses of the radio and sensing vertex are the CAUSEs that we need to infer, as they cannot be observed from outside. The link, path and connection vertices are SYMTONs which we can obtain through probe technique. The data report vertex statues is directly observed at the sink. The Belief Network structure can be easily updated from node tracing module as network topology changes over time.

2) Belief Network inference

When the Belief Network construction completes, the next important issue is to assign CPD to each vertex. The prior fault probability distribution of the radio communication and sensing function are assigned according to experience data. The CPDs of other SYMTON vertices are encoded as noisy-OR gate [21] and Select gate [20]. In a noisy-OR gate, any one of the parent vertices in DOWN status results in the DOWN status of the child variable. In our model, the data report and sensing function vertices affecting the data report vertex, the link and connection vertices affecting the path vertex, two radio vertices affecting the link vertex, and the link and connection vertices affecting path vertex are represents as noisy-OR gate. The relation between multiple paths and a connection is represented as the Select gate, because if any one path is in UP status; the

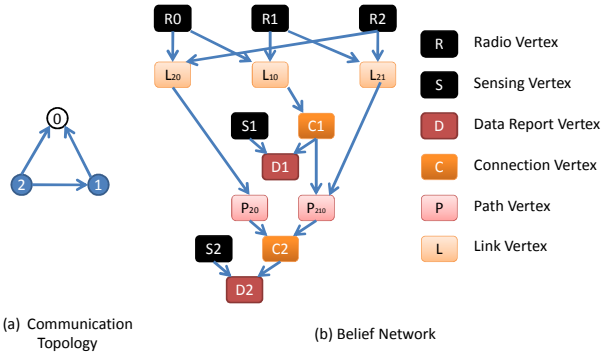


Fig.3. Belief Network. (a) communication topology. (b) belief nework built on communication topology.

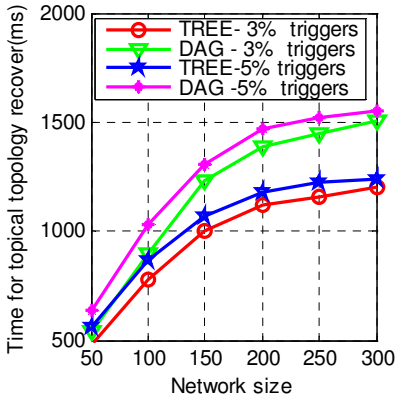


Fig.4. Topical topology reover time on varying network sizes.

connection is in UP status. The contribution of each path to connection vertex can be obtained from the relay lists collected in trace collecting module. The outputs of the inference process are the posterior probabilities about the radio and sensing vertices which provides important guidance for probe selection and root-cause judgment.

B. Probing scheme

Once the probabilistic inference model constructed, we can start our incremental probing process according to information provided by belief network. The main idea of our approach is that if the observed SYMPTOMs are not sufficient to explain the problem, it selects additional optimal probe to explore the most critical SYMPTOMs that are important to explain the problem. There are three major modules: Fault Deducing, Fault Evaluation and Probe Selection. Fault Deducing module puts observed SYMPTOMs into inference model and generates a candidate fault set according to updated posterior probabilities of CAUSEs. Then this set is sent to Faults Evaluation module to compute its belief value and check if it is satisfactory. If the correlated SYMPTOMs are sufficient to explain the fault candidates then probing process terminates. Otherwise, the unobserved symptoms that contribute to the fault candidate set are sent to the Probe Selection module. An optimal probe is selected to discover statuses of those unobserved symptoms. Afterward the probe result is sent to Fault Deducing module for another round of reasoning until a high belief value of the candidate set found or all SYMPTOMs observed.

1) Faults Deducing

The task of fault deducing is to search for root causes of the observed SYMPTOMs. The observed SYMPTOMs sent to fault deducing contain data report statues from the sink and statues of network elements examined by dynamic probing. When new observed SYMPTOMs arrive, the fault deducing module sets those discovered evidences to inference model and update probability of each CAUSE. Then we use a fault probability threshold as a criterion to find which CAUSE is fault. The CAUSE whose newest updated probability of DOWN statues exceeds the threshold is included into candidate fault set. This is a simplified method to find faults which have the most informative contribution to observed SYMPTOMs. Note that, if the threshold is set too high, at initial stage maybe no CAUSE is qualified as a fault. So we can simply choose the CAUSE which has maximum probability of status DOWN as a fault for a start point. It is also noticed that there is no direct relation between two successive deduced candidate fault set because new SYMPTOMs observed may increase or decrease their fault probabilities. As networks statues are progressively probed, more SYMPTOMs are observed which indicate that faults deduced is more closed to ground truth.

2) Faults Evaluation

The candidate fault set generated by the Fault Deducing module maybe inaccurate because the information about the SYMPTOMs statuses is insufficient. The Fault Evaluation module measures the credibility of a candidate fault set given correlated SYMPTOMs and observed SYMPTOMs.

We design a belief value function $BV(C)$ to compute the belief value of given candidate fault set. We assume a candidate fault set consists of a number of faults, denoted as $C_f = \{f_1, f_2, ...\}$. And the SYMPTOMs related to f_i are included in set $S_{fi} = \{SYM_1, SYM_2, ...\}$. So the correlated SYMPTOMs set of a candidate fault set is denoted as $S_c = \bigcup_{fi \in C_f} S_{fi}$. We use S_o to denote observed SYMPTOMs set. Thus, the belief value of a candidate fault set is computed as follows

$$BV(C) = \frac{\prod_{SYMi \in S_r} (1 - \prod_{fi \in C_f} (1 - p(SYMi = DOWN \mid fi))}{\prod_{SYMi \in S_o} (1 - \prod_{fi \in C_f} (1 - p(SYMi = DOWN \mid fi))}$$

3) Probe Selection

If the belief value of a candidate fault set does not exceed the predefined threshold, we should select an optimal probe to gather information about the statues of SYMPTOMs which we most want to know. As stated in [10][13][22], the problem of probe selection whose goal is to minimize probe cost is equal to a minimal set cover problem which is NP-complete.

Therefore, we propose a heuristic greedy algorithm to solve this problem. The main idea is that we always select the probe which covers the maximum number of unobserved SYMPTOMs that related to candidate fault set. If there are several probes which cover the same number of unobserved SYMPTOMs, the probe which explores least number of SYMPTOMs is selected. Our algorithm intuitively appreciates probes that discover more symptom correlation.

VI. EVALUATION

We evaluate DID by extensive simulations and implementation. For the inference model, we use the BNJ implementation of the Belief Network. We implement the node tracing scheme on the TinyOS platform [26] with nesC language and incremental probing scheme with java. The simulation platform is TOSSIM. The sensor nodes in our field experiments are TelosB motes.

A. Simulations

In our simulation, a sensor network is composed of a sink which is located at the center and several nodes which scattered in two-dimensional space. Sensing data is periodically sent to sink through multi-hop communications. Tree-based routing scheme and DAG-based routing scheme are both evaluated. We use 0.5 as the fault posterior probabilities threshold. We also apply the detection ratio and false positive ratio to examine the effectiveness of our incremental probing scheme and inference model.

1) Efficiency of the Node Tracing Scheme

We evaluate topical topology recover time under different network settings. In tests, we simulate a data acquisition sensor network of both TREE routing and DAG routing. Each sensor node generates sensing data and delivers to sink every 1s. Besides the routing schemes, the number of triggers which initiate the trace collecting process also impacts the topology reconstruction. Thus, we evaluate the performance of DID with different number of triggers. Here, 5% triggers means that there are 5% number of nodes which fails to deliver sufficient sensing data to the sink. In tests, we measure the reconstruction time of topical topology. According to the results in Fig. 4, it is very fast that the topical topology recovered by 5% triggers under tree-based routing can be within 1200ms with up to 300 nodes. As nodes in DAG always have multiple parents, the topology of DAG routing is more complicated than it of TREE, so more time is needed to recover it. As the network size increases, the growing speed of topical topology recover time is less than the linearity which indicates a proper scalability of our approach.

2) The Performance of incremental probing

We then evaluate the performance of incremental probing with four different groups of simulations. We inject artificial errors into the network and use incremental probing scheme to refine the inference model which generates diagnosis reports according to the posterior probability estimations.

Firstly, only sensing failures are inject into sensor nodes. We randomly set the sensing function disabled in 5% of the nodes. The network size varies from 50 to 100 nodes. As the detection ratios shown in Fig. 5(a), our approach achieves detection ratios higher than 90% in both TREE and DAG settings. Fig. 5(b) describes the false positive ratio. We see that the false positive ratio decreases as the network size increases. Through analyzing the intermediate and final diagnosis reports and, this can be explained by that the bigger network size, the more link-related SYMPTOMs are easily observed, and so the higher accuracy of reasoning sensing failure. Otherwise, the posterior probability of sensing failure is heavily relied on the prior fault probability and improper faults threshold might result in negative judgment.

We then randomly inject sensing failure and radio communication failure into sensor nodes. The result is shown in Fig. 6. Both decrease rate of the detection ratio and increase ratio of the false positive ratio is relatively high when the number of nodes is below 60. After the network size grows beyond 60, the detection ratio and false positive ratio are both relatively stable. This is mainly because the small size of belief network always performs better than the large size in probabilistic inference.

Another group of simulation evaluates different belief value threshold of candidate fault set. The experiment is conducted under TREE routing setting. The Other settings are the same as previous group simulation. As the experiment result shown in Fig. 7, both thresholds achieve detection ratios higher than 75%. We also can find that the higher of threshold value, the lower of detection ratio and false positive ratio. This phenomenon is consistent with the intention of designing belief value function to evaluate the quality of candidate faults.

In the last group of simulations, we fix the network size at 30 and examine the effect of simultaneous multiple faults in our model which vary from 1 to 5. As shown in Fig. 8, as more faults occur, the detection ratio decreases and the false positive ratio increases. This is the result of that mutual interference of multiple faults in the inference model deteriorate inference performance.

B. Field Experiment

We also implement and evaluate the effectiveness of the DID approach in our forest monitoring sensor networks [23]. We conduct the experiment over a period of more than three months, and we select and analyze a six-day data trace that consists of a segment of 42228 received packets from 309 sensor nodes.

1) The Trace Diagnosis

Our trace analysis mainly consists of three parts. First, we count the sensing data receptions in *TIMEPERIOD1*, if some nodes did not sufficiently hand in their sensing data, we call them suspicious nodes. Then we start the probing diagnosis in *TIMEPERIOD2*. The topical topology is constructed based on the neighbor list of nodes which are involved in the relaying

paths of suspicious nodes during *TIMEPERIOD1*. The probing process is equal to searching the correspondent paths in the *TIMEPERIOD2*. According to the incremental probing results, the inference model finally deduces the sensing failures and node failures. In Table I, we list the diagnosis reports of two time periods in our six-day experiment. In diagnosis report

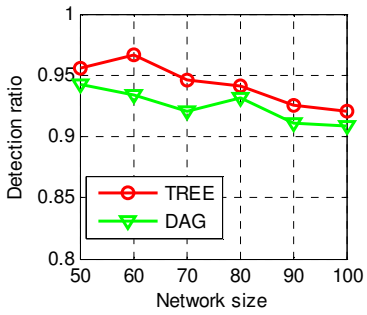


Fig.5. (a) Sensing failure detection ratio

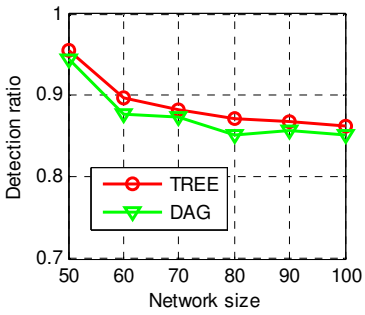


Fig.6. (a) Sensing and radio failure detection ratio

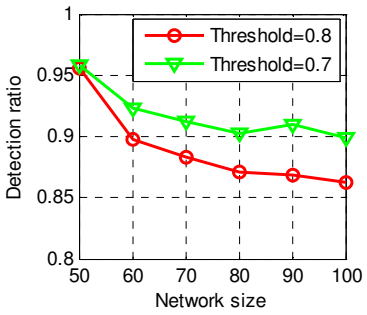


Fig.7. (a) Detection ratio for varied threshold

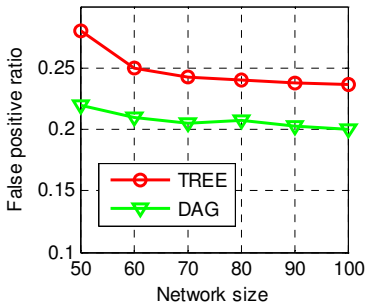


Fig.5. (b) Sensing failure false positive ratio

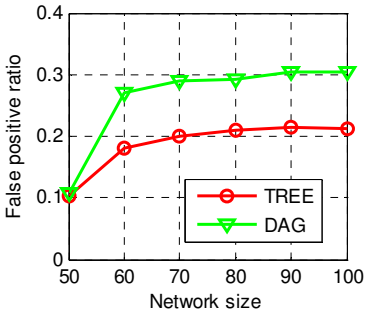


Fig.6. (b) Sensing and radio failure false positive ratio

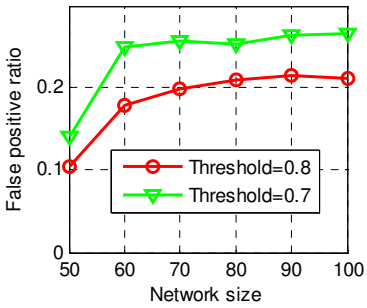


Fig.7. (b) False positive ratio for varied threshold

I, through 21 incremental probes, we localize 12 sensing failures and 7 radio failures of nodes while we localize 15 sensing failures and 9 radio failures of nodes using 25 probes in diagnosis report II.

TABLE I. DIAGNOSIS REPORTS

	Diagnosis report I	Diagnosis report II
Sensing failures	12	15
Radio communication failures	7	9
Number of probes	21	25

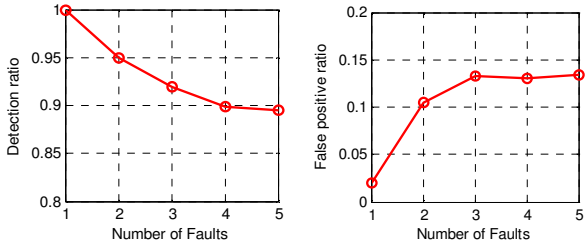
2) Faulty Network Diagnosis

In this experiment, we artificially inject sensing failures and radio failures into the network with 30 sensors. As shown in Fig.9, the traffic overhead incurred by DID is very low. We can see that diagnosis overhead is no more than 15% of the total traffic in most situations. As the number of faults increases, the growing speed of diagnosis overhead is less than the linearity, showing the scalability of our approach. Note that this overhead is introduced only when the diagnosis process is launching. The time of diagnosis is comparatively small to long-term normal running of WSNs, so this overhead is truly acceptable.

Then, we inject one sensing failure node (NodeA) and one radio failure node (NodeB) into the network. As shown in Fig. 10, the variations of inferred posterior fault probability of both faulty nodes are given. As probes incrementing, more SYMPTOMs are effectively observed. After each probe, the fault probabilities are gradually close to the ground truth which indicates that NodeA and NodeB are both faulty.

VII. CONCLUSION

We propose DID, a directional diagnosis approach which can be easily implemented for providing online diagnosis of working WSNs. The incremental probing scheme enables the diagnosis information collecting guided by the updated inference model which can quickly localize faults as well as minimize communication cost incurred by collecting useless information. The proposed node tracing scheme can successfully recover topical topology for diagnosis with relatively low overhead. We implement our approach and examine its effectiveness in our forest monitoring project. The diagnosis result is accurate in real-time. We now plan to introduce a decentralized mechanism for diagnosis information collecting, so that the sink does not have to maintain the most updated state of all nodes in the network. In addition, more network exceptions and unexpected faults such as environmental interference are considered to be included into inference model for further reasoning.



(a) (b)

Fig.8. (a) Detection ratio for multiple faults (b) False positive ratio for multiple faults.

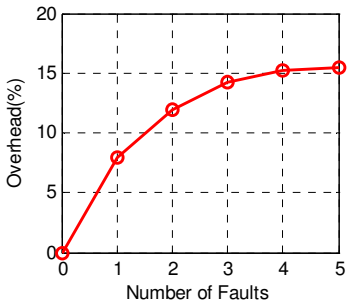


Fig.9. Diagnosis overhead in DID.

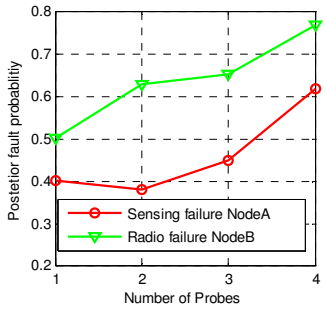


Fig.10. The postetior probablitiy variations.

ACKNOWLEDGEMENTS

This work is supported in part by the 973 Program of China (Grants No. 2010CB328000 and No. 2011CB302705), the National Natural Science Foundation of China (Grant No. 61073168), the NSFC/RGC Joint Research Scheme N_HKUST602/08.

REFERENCES

[1] R. Szewczyk, A.Mainwaring, J. Anderson, and D. Culler. “An Analysis of a Large Scale Habit Monitoring Application”. In SenSys’04, 2004.

[2] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. “A Wireless Sensor Network for Structural Monitoring”. In SenSys’04, 2004.

[3] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, “Energy-efficient surveillance system using wireless sensor networks,” in Proc. ACM MobiSys, 2004, pp. 270-283.

[4] O. Chipara, C. Lu, T.C. Bailey and G.-C. Roman, “Reliable Clinical Monitoring using Wireless Sensor Networks: Experience in a Step-down Hospital Unit”, In SenSys'10, November 2010.

[5] G. Hackmann, O. Chipara and C. Lu, “Robust Topology Control for Indoor Wireless Sensor Networks”, In SenSys'08, November 2008

[6] O. Chipara, G. Hackmann, C. Lu, W. Smart and G.C. Roman, “Practical Modeling and Prediction of Radio Coverage of Indoor Sensor Networks”, In IPSN'10, April 2010.

[7] S. Lim, C. Yu, and C. R. Das, “Rcast: A randomized communication scheme for improving energy efficiency in MANETs,” in Proc. IEEE ICDCS, 2005, pp. 123-132.

[8] M.M.H.Khan, H.K.Le, M.LeMay, P.Moinzadeh, L.Wang,Y.Yang, D.K.Noh T.Abdelzaher, C.A.Gunter, J.Han and X.Jin, “Diagnostic Powertracing for Sensor Node Failure Analysis,” in Proc. IEEE IPSN,2010, pp.117-128.

[9] J. Zhao, R. Govindan, and D. Estrin, “Residual energy scan for monitoring sensor networks,” in Proc. IEEE WCNC, 2002, vol. 1, pp. 356-362.

[10] Y. Bejerano and R. Rastogi, “Robust Monitoring of Link Delays and Faults in IP Networks,” in Proceedings of IEEE INFOCOM ’03, April 2003.

[11] N. Duffield, “Network tomography of binary network performance characteristics,” IEEE Transactions on Information Theory, vol. 52, pp. 5373-5388, 2006

[12] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, “NetDiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data,” in Proceedings of ACM CoNEXT ’07, December 2007.

[13] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren, “IP fault localization via risk modeling,” in Proc. USENIX NSDI, 2005, pp. 57-70.

[14] S. Kandula, D. Katabi, and J.-P. Vasseur, “Shrink: A tool for failure diagnosis in IP networks,” in Proc. MineNet, 2005, pp. 173-178.

[15] G. Tolle and D. Culler, “Design of an application-cooperative management system for wireless sensor networks,” in *EWSN*, January 2005.

[16] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin, “Sympathy for the sensor network debugger,” in *SenSys*, November 2005.

[17] L. Cheng, X. Qiu, L. Meng, Y Qiao, and R. Boutaba, “Efficient Active Probing for Fault Diagnosis in Large Scale and Noisy Networks”, In Infocom’10, pp. 2963-2973

[18] K. Liu, M. Li, Y. Liu, M. Li, Z. Guo, and F. Hong, “Passive diagnosis for wireless sensor networks,” in ACM International Conference on Embedded Networked Sensor Systems, 2008, pp. 113-126.

[19] 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.

[20] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang, “Towards highly reliable enterprise network services via inference of multi-level dependencies,” in Proc. ACM SIGCOMM, 2007, pp. 13-24.

[21] J. Pearl, “Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference”, San Mateo, CA: Morgan Kaufmann, 1988.

[22] I. Rish, M. Brodie, N. Odintsova, S. Ma, and G. Grabarnik, “Real-time Problem Determination in Distributed Systems Using Active Probing,” in Proceedings of the 9th IFIP/IEEE International Network Management and Operations Symposium (NOMS 2004), Seoul, Korea, April 2004, pp. 133-146.

[23] GreenOrbs, http://www.greenorbs.org/

[24] X. Mao, X. Li, X. Shen, F. Chen. “iLight: device-free passive tracking by wireless sensor networks”. In Proceedings of the 7th International Conference on Embedded networked sensor systems (SenSys’ 09), pp. 315–316.

[25] Xufei Mao, S. Tang, X. Li and Y. Sun. “MENs: Multi-user Emergency Navigation System Using Wireless Sensor Networks,” in Ad Hoc & Sensor Wireless Networks, 2010

[26] TinyOS, http://www.tinyos.net/