

ADAPTIVE FAULT LOCALIZATION IN MOBILE AD HOC BATTLEFIELD NETWORKS*

Maitreya Natu, Adarshpal S. Sethi
University of Delaware, Newark, DE, 19711
{natu, sethi}@cis.udel.edu

ABSTRACT

Future battlefield networks require extensive fault management mechanism. Most existing fault localization algorithms assume availability of complete and/or deterministic dependency model. Such assumptions can not be made in the dynamic environment of a battlefield network where nodes may move, causing periodic changes in routes. This paper is aimed at developing a fault diagnosis architecture and algorithm, which address the issue of dynamically changing dependencies in battlefield networks.

1. INTRODUCTION

Battlefield networks require an extensive fault localization mechanism to provide robust networking services in the presence of random failures. This demand becomes more critical in the Army's future battlefield networks to meet the challenge of service survivability which requires efficient and accurate fault localization algorithms to isolate the root cause of failure. The failure information can then be used to perform healing measures for providing uninterrupted services in the ad-hoc battlefield environment [1, 14]. The task of fault localization [5] is complex and challenging. The challenges become more critical in a battlefield environment because of the following reasons:

- Battlefield networks are usually ad-hoc networks, thus no static infrastructure can be assumed in the design,
- A higher degree of non-determinism is present in the environment because of dynamically changing topology and lack of accurate and timely information,
- Large array of possible reasons can explain an observed behavior, e.g., soft and hard failures, transient and non-transient failures, failures due to hostile or unintentional attacks, etc.,
- Various battlefield applications have varying levels of tolerance in delays and losses, e.g., mission critical applications require quick and assured delivery while non mission critical applications are tolerant to some delays and losses. Thus the fault localization mechanism needs to balance the performance and survivability requirements of various applications.

Many algorithms have been developed in the past [4, 11, 12, 13] for localizing a fault through correlation of alarms or symptoms observed in the network. These algorithms vary in the network models used, complexity involved in computation, the assumption made about the underlying network etc. Recently a promising new algorithm called Incremental Hypothesis Updating (IHU) [2] was designed which processes symptoms one at a time in an incremental fashion, thereby providing increased efficiency. As part of a multi-year research task under the Army Research Lab (ARL) Collaborative Technology Alliance (CTA) program, we have designed and implemented the IHU algorithm and provided a set of preliminary results on its performance. The IHU algorithm has been shown to be fast, scalable, and accurate with the potential of being deployable in real-time. However a number of issues arise in using this algorithm in wireless ad-hoc battlefield networks. An important issue that needs to be addressed in such an environment is the presence of mobility in the managed network. Most of the existing fault diagnosis methodologies [1, 6] assume availability of a complete and deterministic dependency model. This assumption can not be made in battlefield networks, as the nodes may not be static and thus the topology may keep changing with time [10]. In this scenario, the fault localization algorithm needs to adapt to the changing dependencies.

This paper is aimed at developing a fault localization algorithm that takes into account the dynamically changing dependencies in battlefield networks. The paper has three main contributions. First, it presents models to incorporate temporal information to associate time with fault-symptom relationships, and with each reported symptom. This time information indicates the relevance of the dependency model for processing a particular symptom. The second contribution is a fault correlation algorithm that adapts to the dynamically changing dependencies. The algorithm uses the dependency model to process the observed symptoms incrementally as they are received, and modifies the hypothesis on receiving the changed topology information. The algorithm reports a set of hypothesis as possible causes of the reported failure symptoms and ranks them based on the degree of confidence that the hypothesis is the possible root cause of failure. These ranks (beliefs) are computed based on the probabilistic dependency model and the temporal information present in the dependency

* Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

model, and the reported symptom. As a the third contribution, in the paper we present simulation results done in Qualnet to show the correctness and efficiency of the algorithm to be deployed in real time battlefield networks.

2. ADAPTIVE APPROACH TO FAULT LOCALIZATION IN BATTLEFIELD NETWORKS

Battlefield networks, being wireless, mobile, and ad-hoc, possess many unique characteristics [7] because of which existing techniques of fault localization can not be directly used. Many fault localization algorithms depend on the dependency model [6] for fault localization. This model represents the causal relationships between faults and symptoms. Since causal relationships between faults and symptoms are difficult to determine, the dependency model is a probabilistic one, in which the fault nodes are associated with the probability of its independent occurrence, and the fault-symptom dependencies are represented with the probability of the causal implication between faults and symptoms. Accuracy of fault diagnosis relies upon the accuracy of the dependency model. Thus building a complete, consistent, and accurate dependency model is a critical task. Prior problem determination work assumes a prior knowledge of symptom-fault dependencies. However this requirement is not met completely and hence periodic update of the dependency model is required in most real-time situations for several reasons:

- Node mobility: In wireless ad-hoc networks, node mobility makes the topology dynamic and unpredictable. Due to mobility, nodes and links in the managed network continually get added, removed or reconfigured. These dynamic changes affect the symptom-fault dependencies.
- Incomplete and inaccurate dependency information: Practical tools and techniques used to construct the dependency model may fail to discover all the existing dependencies. Moreover the gathered dependency information might not be accurate. For instance, the problem of detecting end-to-end routes is aggravated by the presence of load balancers, multi-path routes, and dynamic route changes. The routes discovered in such scenarios might keep changing and hence provide partial or inaccurate information about the existing symptom-fault dependencies.
- Gradual increase in the preciseness of the belief: With time, as more and more problem determination tasks are performed and the root-cause analysis is done, the strengths of the causal influences between symptoms and faults, represented by forward conditional probabilities, can be changed to more precise values.

A big challenge is to periodically update the model in parallel with the ongoing fault diagnosis process. The fault

diagnosis process extracts information from the dependency model to perform event correlation. Thus in presence of multiple dependency information, determining which dependency information to associate with the observed symptom is a challenging task. An appropriate data structure must be designed to represent the dependency model to incorporate:

- Changes in the fault set and symptom set that would arise when network elements are added or removed, resulting in the addition or deletion of faults and symptoms.
- Changes in the fault-symptom relationships, which may occur when network topology changes. This can cause existing relationships to become obsolete. It may also add new relationships.
- Changes in the probabilities associated with the fault-symptom relationships, which may occur over time as a result of the learning process.

We propose to develop an adaptive component to incorporate dynamically changing dependencies in the dependency model and timestamp this information to perform temporal correlation of the symptoms.

Event driven techniques [6] maintain a state which encodes partial fault-localization results computed on the basis of the observed symptoms. Symptoms are analyzed independent of the other symptoms and the fault localization results are updated accordingly. However, by updating the partial fault-localization information with each arrived symptom, the information about the individual symptoms is lost.

Maintaining the information about the sequence of individual symptoms that contribute to the overall fault diagnosis can help to refine the algorithm results. We propose to build a *hypothesis search space (HSS)* to keep track of the reported symptoms, their associated data, and their effect on the hypothesis construction. This model represents the observed symptoms and their dependency relationship with the possible faults. Maintaining this data structure, instead of incrementally updating the hypothesis with each observed symptom, can help in various ways to refine the collected information about the network health.

- Removing spurious symptoms: The observation of network behavior is frequently disturbed by the presence of spurious symptoms [2]. These are caused by transient network failures or due to overly restrictive threshold values. Considering these symptoms in the fault localization algorithm can reduce the accuracy of the algorithm. A symptom can be detected to be spurious if it is reported for small time duration. A spurious symptom can be removed from the HSS if a healthy re-

sponse for the same probe is obtained within a certain time frame [3]. The length of the time frame can be decided on the basis of the frequency of probing, response time, and other parameters.

- Incorporating the updated dependency information: As the probability values of the causal relationship between the symptoms and faults are updated, this information can also be incorporated in the HSS by updating the probabilities representing the symptom-fault causal relationship.

Because of the changed design of the dynamic dependency model, the incremental algorithm needs modification. The algorithm must be modified to deal with dynamic addition and deletion of symptoms, and faults in parallel with ongoing hypothesis construction. It is also necessary to properly deal with symptoms that arise before and after such changes to the dependency model have occurred. One method of solving this problem is to incorporate temporal information into the dependency model and the incremental algorithm.

3. SYSTEM ARCHITECTURE

Figure 1 shows the different modules cooperating to perform adaptive fault localization. The topology discovery agents periodically report the topology updates. These updates are incorporated into the dependency model by the adaptive component. Network monitors probe the network components and report the observed symptoms. These symptoms are stored in the Hypothesis Search Space (HSS). The network manager performs symptom-fault correlation and temporal correlation of these symptoms to provide the most probable explanation.

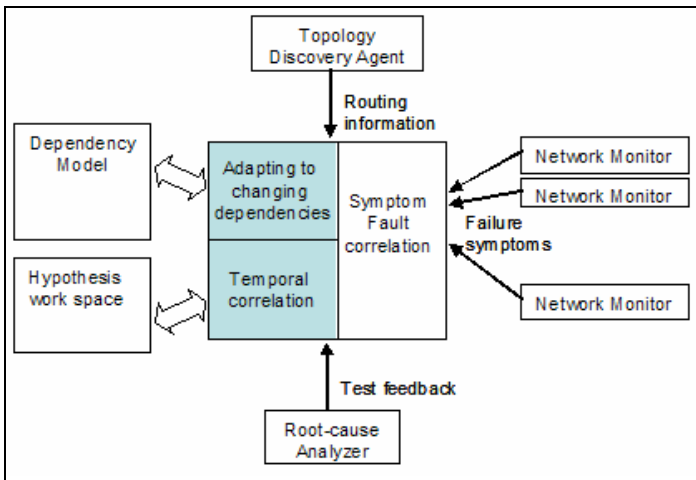


Figure 1. System Architecture

3.1 Building the dependency model

The dependency model is implemented as a matrix with rows representing the observable symptoms and the columns representing the possible network faults. Each matrix cell (i, j) has a probability value [6] representing likelihood of the cause-effect implication between the i^{th} symptom and j^{th} fault. In our experiments we model end-to-end path failures as symptoms and the link failures as faults. The dependency information is computed based on the route information provided by the topology discovery agents. The routes obtained by the topology discovery agents indicate the possible links that can cause the end-to-end path failure. The network manager uses this information to build the dependency matrix. The probability values are initialized to some constant value and are updated with time. As discussed earlier, the obtained dependency information can be incomplete or inconsistent. Hence, in the presence of incomplete and changing dependency model, it is essential for the network manager to have an adaptive component. This component will constantly update the dependency model if it discovers any changes in the dependencies, e.g., due to dynamic routing, addition/deletion of nodes etc. These topology changes are reported to the manager by the topology discovery agents. The changes are included into the model by updating the symptom-fault probability values. Addition/deletion of new nodes is done by adding/removing new rows and columns in the matrix. Another role of the adaptive component is to refine the probability values. These values are initially set to some constant value and are increased or decreased with time, based on the level of confidence in the obtained dependency information. The results of the root cause analyzer are used to refine these probability values.

3.2 Building the hypothesis search space

The hypothesis search space (HSS) is also implemented as a matrix representing the symptom-fault relationship, but unlike the dependency model which contains information of all possible symptoms, HSS stores only the symptoms observed in a time window. On arrival of each symptom reported by the network monitors, the symptom is searched in the dependency model and a new row is added to the HSS indicating the symptom-fault dependencies between the symptom and its possible causes of failure. Thus this data structure changes more frequently than the dependency model because of frequent addition/removal of symptoms. Maintaining this data structure allows removing symptoms that are too old and thus irrelevant to the current fault diagnosis process by maintaining a time window. It allows another task of updating the probability values with changes made in the dependency model. The event correlation algorithm described in the next section assigns temporal weights to the dependencies that explain the arriving symptoms.

4. INCORPORATING TEMPORAL INFORMATION IN FAULT LOCALIZATION

The existing incremental algorithm does not have a notion of time while performing event correlation. In the dynamically changing environment of a battlefield network, it becomes important to incorporate temporal information to improve the accuracy of the fault diagnosis. Temporal information can be introduced at following places in the architecture:

- Associating time with fault-symptom relationships: Due to change in the topology the fault-symptom relationships might change with time. Thus these changing relationships might provide different explanations for the observed symptoms. These relationships need to be time stamped to judge the relevance of dependencies while processing a symptom which arrived at a particular point in time.
- Associating symptom with the symptom-arrival time: Putting a timestamp with each symptom allows the manager to infer an approximate time when the symptom was generated. This helps the manager to infer the relevance of symptom fault dependencies for processing the symptom. Another advantage of associating time with each symptom is to identify when a symptom gets too old to be discarded from the fault diagnosis.
- Associating time with a hypothesis: Hypothesis sets are built with the arrival of each new symptom. If a hypothesis is too old then it is less likely to explain the current network situation. Thus discarding the old hypothesis might allow the fault diagnosis algorithm to converge better to identify the root cause. Time-stamping the hypothesis sets allows the manager to identify the relevance of hypothesis.

4.1 The event correlation process

In this section we propose an adaptive approach to the event correlation algorithm. The approach presented in this paper uses the belief computation technique presented in the IHU algorithm [2] and extends it to incorporate temporal correlation and dynamic dependency changes. The manager correlates the observed symptoms reported by the network monitors and creates a set of the most likely hypotheses explaining the set of observed symptoms. It creates the hypotheses sets on a continuous basis and incrementally updates them with the information learned from the arriving symptoms. The hypotheses are ranked using a belief metric, which expresses a relative confidence associated with the hypothesis with respect to other hypotheses. In other words a belief represents relative importance of the hypothesis.

The algorithm proceeds in an incremental fashion, with arrival of each new symptom. To incorporate explanation

for a symptom S_i into hypothesis, faults explaining S_i are incorporated in the HSS. This is done by adding a new row in the HSS from the dependency model, which explains the observed symptoms. With each new symptom the hypotheses are modified to incorporate explanation of the new symptom.

As the dependency model is updated periodically, the arriving symptoms can be explained by the dependency model built before the symptom arrival or the one that is built after symptom arrival. The relevance of the two explanations can be weighed on the basis of temporal closeness between the build-time of the models with symptom arrival time. Each dependency model can be marked with the timestamp of creation and similarly each symptom can be marked with a timestamp of arrival. The inferred fault-symptom dependencies can be weighed based on the difference in timestamp of the symptom and the model. The smaller the difference, higher is the relevance. Another thing to note is that the symptoms and dependency updates are not reported at the exact time of occurrence. Thus there exists some degree of uncertainty in the temporal information. This needs to be modeled with appropriate probability values while performing temporal correlation.

One approach to this scheme could be to build the hypothesis incrementally on receiving each new symptom, and then rebuild the hypothesis after obtaining the new dependency information. But this approach involves an overhead of redoing the entire hypothesis building process. On the other hand, deferring the hypothesis generation till the next topology update might cause unnecessary delay in producing the hypothesis if the topology update occurs a long time after the symptom is reported. This also reduces its probability of being relevant to the reported symptom. This approach does not make use of the capability of the Incremental algorithm to perform fault correlation in the inter-symptom arrival time.

To meet this trade-off, the symptom processing is deferred for a certain time window. If no topology update is received within this time window, the hypothesis is built for the symptom based on the available dependency information. If a topology update arrives within the time window, the dependency information from both the available and the new topology information can be temporally weighted and used for generating the explanation for the symptom. This avoids the overhead of rebuilding the entire hypothesis. The size of the time window can be decided based on the nature of the network. If the network topology changes periodically, and the topology updates are frequent, the time-window can be set to the time between arrivals of two topology updates. On the other hand, if the network is relatively static and the topology updates are relatively infrequent, the time-window need not be as large as the time

between two topology updates. It can be set to some minimum time for a change to be reported to the manager.

On processing of each new symptom, a new row is created in the HSS. If no topology update is received within the time window of this symptom, row contains dependency information obtained from the present dependency model. If however, a topology update is received within the symptom's time-window, row contains dependency information obtained for that symptom from both the dependency models. The dependency probabilities are weighted with the difference in the timestamp of the symptom and the dependency model. This process thus takes care of adapting the event correlation process with the continuously changing dependencies.

An important task is to compute beliefs for each hypothesis. As described in the IHU algorithm [2], beliefs can be computed as:

$$b_i(h_j) = \beta \prod_{f_k \in h_j} p(f_k) \prod_{s_l \in S_{O,i}} (1 - \prod_{f_k \in h_j} (1 - p(s_l|f_k)))$$

On receiving a symptom s_i out of the possible observable symptoms $S_{O,i}$, a hypothesis set H_i is created. H_i is a set of hypotheses h_j . Each hypothesis h_j is a collection of faults f_k that explain the occurrence of symptoms s_1 through s_i . Set H_i is created by updating H_{i-1} with an explanation of symptom s_i .

In the i^{th} iteration of fault localization, the belief metric $b_i(h_j)$ is expressed using the probability that (1) all faults belonging to $h_j \in H_i$ have occurred, and (2) h_i explains every observed symptom $s_k \in S_{O,i} = s_1, \dots, s_i$

The belief values can be refined further by the fact that some possible indications of disorder have not been observed. This lack of observation is considered a positive symptom. Thus the belief metric needs to contain a negative component and a positive component such that:

$$b_i(h_j) = \beta b_i^p(h_j) b_i^n(h_j)$$

The negative component is the same as the belief computed previously. The positive component is defined as the probability that faults in h_j have not generated any of the symptoms in $S_{O,i}$. This probability is expressed through the following equation:

$$b_i^p(h_j) = \prod_{s_l \in S_{O,i}^s} \prod_{f_k \in h_j} (1 - p(s_l|f_k))$$

This multiplier decreases the value of the belief metric associated with the hypothesis if many of the symptoms caused by the faults in h_j have not been observed.

As we are storing the individual symptom-fault dependencies in the HSS, the above belief computations need not be done after arrival of each new symptom. Instead, the beliefs can be computed after collecting a set of symptoms or after a window of time has elapsed. However the hypothesis sets need to be created with each new symptom arrival as these sets are built incrementally. Negative belief computation for a hypothesis set H can be done by fetching the conditional probability values between the faults in a hypothesis set and all the observed symptoms. Positive belief for a hypothesis set can be computed by obtaining the conditional probability values between the faults in a hypothesis set and all the unobserved symptoms. The positive belief computation involves lesser computation than when done incrementally for each arriving symptom.

5. EXPERIMENTS AND RESULTS

In this section we evaluate the technique presented in this paper using the problem of end-to-end connectivity failure diagnosis as a case study. We deal with the problem of isolating interface failures responsible for an end-to-end connection failure in mobile ad-hoc networks. In this problem, the dependency matrix consists of end-to-end connection failure as rows which are reported as symptoms. The columns consist of interface failures that represent the possible faults. In an n -node network, there can be at most n^2 end-to-end connections, each composed of at most n hop-to-hop services. We obtained the dependency information by periodically running trace-routes. End-to-end connection failure information is obtained by periodically sending pings to all the managed nodes in the network.

We simulated networks of different sizes in Qualnet [9] and introduced interface failures for random time interval. To observe the effect of mobility and transmission power, we used random waypoint model and observed the accuracy of detection by varying the average speed and transmission power.

Figure 2 shows the relation between the detection rate and transmission power for a network with 15 nodes varying the nodes' average speeds from 3m/s to 11m/s, and transmission power from 11dBm to 20dBm. We observed that detection rate decreases for higher speeds. Figure 3 gives a better insight for this observation, where detection rate tends to decrease with increase in speed, for all values of transmission power. The decrease in detection rate with increased mobility can be explained by the lack of accuracy of the dependency information inferred about the network. With increase in speed, nodes change position faster causing changes in routes. These changes are not accurately retrieved by the traceroutes, causing an inaccuracy in the fault diagnosis.

6. CONCLUSION AND FUTURE WORK

In this paper, we described a strategy for fault diagnosis in a dynamically changing environment of mobile ad-hoc battlefield networks. We designed a system architecture that adapts to changing dependencies in the network. We illustrated a method to perform temporal correlation and proposed an adaptive fault diagnosis algorithm. We evaluated the proposed method by simulations in Qualnet and observed the effect of mobility and transmission power on the detection rate of the fault diagnosis algorithm. We aim to conduct more detailed simulations of battlefield networks to understand the impact of mobility. Additionally we also aim to test the algorithm's performance on network with larger sizes. The experiments presented in this paper have a static time window size. As a continuing work, we will incorporate mechanism to adapt the window size to the observed degree of mobility.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied of the Army Research Laboratory or the U.S. Government.

References

- [1] L. Kant, W. Chen, C-W. Lee, A. S. Sethi, M. Natu, L. Luo, and C-C. Shen. D-FLASH: Dynamic Fault Localization and Self-Healing for Battlefield Networks. In *ASC'04, the 24th Army Science Conference, Orlando, FL*, Nov.-Dec. 2004.
- [2] M. Steinder and A. S. Sethi. Probabilistic fault diagnosis in communication systems through incremental hypothesis updating. *Computer Networks*, 45(4):537–562, July 2004.
- [3] S.A. Yemini, S. Kliger, E. Mozes, Y. Yemini, D. Ohsie, High speed and robust event correlation, *IEEE Communications Magazine* 34 (5) (1996) 82–90.
- [4] M. Steinder, Fault localization in communication networks—a survey, *Technical Report 2001-01, CIS Department, University of Delaware*, February 2001.
- [5] G. Jakobson, M.D. Weissman, Alarm correlation, *IEEE Network* 7 (6) (1993) 52–59.
- [6] I. Katzela, M. Schwartz, Schemes for fault identification in communication networks, *IEEE/ACM Transactions on Networking* 3(6) (1995) 733–764.
- [7] L. Qiu, P. Bahl, A. Rao, and L. Zhou. Fault Detection, Isolation, and Diagnosis in Multihop Wireless Networks. *Technical Report MSR-TR-2004-11, Microsoft Research, Redmond, WA*, Dec 2003.
- [8] A.T. Bouloutas, S. Calo, A. Finkel, Alarm correlation and fault identification in communication networks, *IEEE Transactions on Communications* 42 (2/3/4) (1994) 523–533.

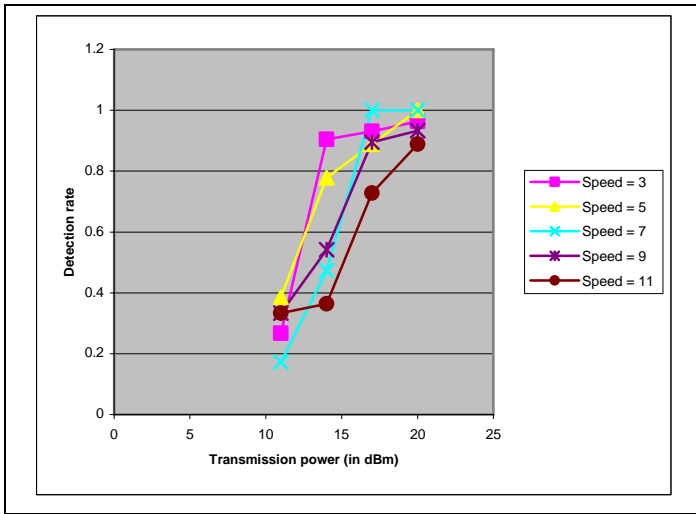


Figure 2. Detection rate vs. transmission power for different values of speed

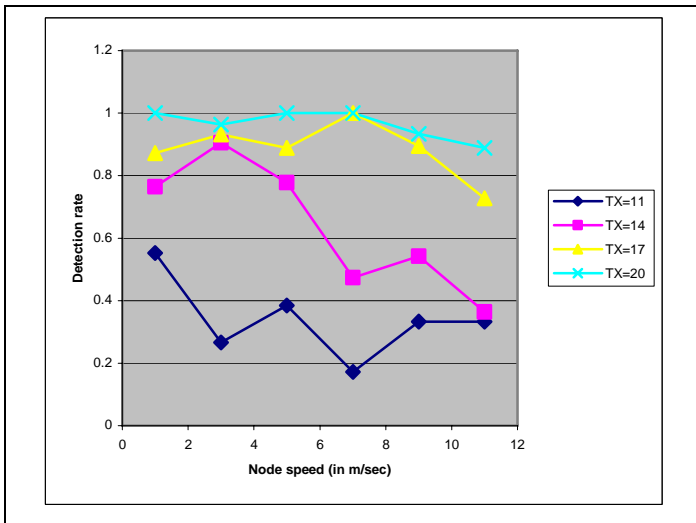


Figure 3. Detection rate vs. speed for different values of transmission power

We also conducted experiments to observe the effect of transmission power on fault diagnosis and we observed that the detection rate improves with increase in transmission power. For low transmission power, low detection rate can be accounted to the inability of the node to report symptoms to network manager and inability of the network manager in extracting complete routing information resulting in incomplete dependency model. We also observed that detection rate reaches higher values beyond a cut-off point. With smaller speeds this cutoff point is reached for smaller transmission power while for higher speeds it takes higher transmission power to reach such cut-off point. E.g. In figure 2 for speed of 3m/s, cutoff is reached at transmission power of 14dBm while for speed of 11m/s, it is reached at transmission power of 17 dBm.

- [9] The Qualnet simulator from Scalable Networks Inc.
<http://www.scalable-networks.com/>.
- [10] D. Kotz, C. Newport, C. Elliott, The Mistaken axioms of wireless-network research, *Dartmouth College Computer Science Technical Report, TR2003-67*
- [11] C. S. Chao, D. L. Yang, and A. C. Liu, "An automated fault diagnosis system using hierarchical reasoning and alarm correlation," *J. Network Syst. Manage.*, vol. 9, no. 2, pp. 183–202, 2001.
- [12] M. Hasan, B. Sugla, and R. Viswanathan, "A conceptual framework for network management event correlation and filtering systems," in *Integrated Network Manage. VI*, M. Sloman, S. Mazumdar, and E. Lupu, Eds., 1999, pp. 233–246.
- [13] M. Steinder and A.S. Sethi, "Non-deterministic Fault Localization in Communication Systems Using Belief Networks." *IEEE/ACM Transactions on Networking* Vol. 12, 5 (Oct. 2004), pp. 809-822.
- [14] L. Kant, A.S. Sethi, and M. Steinder, "Fault Localization and Self-Healing Mechanisms for FCS Networks." *Proc. 23rd Army Science Conference, Orlando, FL* (Dec. 2002).