

Using Temporal Correlation for Fault Localization in Dynamically Changing Networks¹

Maitreya Natu

Dept. of Computer and Information Science

University of Delaware

Newark, DE, USA, 19716

Email: natu@cis.udel.edu

Adarshpal S. Sethi

Dept. of Computer and Information Science

University of Delaware

Newark, DE, USA, 19716

Email: sethi@cis.udel.edu

Abstract

A mobile ad-hoc network creates a dynamic environment where node mobility can cause periodic changes in routes. Most existing fault localization algorithms assume availability of complete and/or deterministic dependency model. Such assumptions can not be made in the dynamically changing networks. This paper is aimed at developing a fault diagnosis architecture and algorithm to address the issue of dynamically changing dependencies in networks. We propose an architecture to capture the changes in the dependencies and introduce a temporal correlation algorithm to perform fault diagnosis with the dynamically changing dependency information. We present an experimental evaluation of our work through simulation results using Qualnet.

I. INTRODUCTION

Many mobile ad-hoc networks demand an extensive fault localization mechanism to meet the challenge of service survivability in the presence of random failures. The problem becomes more challenging with the dynamic nature of the environment. The non-determinism and lack of static infrastructure demand a robust architectural design for localizing the root cause of failure. The failure information can then be used to perform healing measures for providing uninterrupted services [4], [5]. The task of fault localization [3] is complex and challenging. The challenges become more critical in a dynamic environment because of the following reasons:

- With the ad-hoc nature of the networks, no static infrastructure can be assumed in the design.
- Node mobility brings a high degree of non-determinism in the environment because of dynamically changing topology and lack of accurate and timely information.
- Large array of possible reasons for an observed behavior, e.g., soft and hard failures, transient and non-transient failures, failures due to hostile or unintentional attacks, etc.
- Various 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.

¹Prepared 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.

Many algorithms have been developed in the past [3], [14], [1], [2], [7], [9] 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) [13] 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 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 [4], [14], [6] assume availability of a complete and deterministic dependency model. This assumption can not be made in mobile ad-hoc networks, as the nodes may not be static and thus the topology may keep changing with time [8]. In this scenario, the fault localization algorithm needs to adapt to the changing dependencies.

We introduced the concept of adaptive fault localization in [10]. This paper is an extended version of [10] and discusses at length the development of fault localization architecture and a temporal correlation based adaptive algorithm that takes into account the dynamically changing dependencies in networks. The paper has three main contributions.

- In order to capture the dynamic nature of the environment, we present a system architecture to incorporate the dynamically changing dependencies. This architecture introduces a dynamic dependency model to capture changing dependencies and a hypotheses search space to maintain and update the inferred network state. This architecture associates time with symptom-fault relationships, and with each reported symptom. Time 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. We present a temporal correlation algorithm to compute the relevance of dependency information to provide explanation for an observed symptom. The algorithm reports a set of hypotheses 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 model and the reported symptom.
- As the third contribution, in the paper we present simulation results done in Qualnet to show the correctness and efficiency of the algorithm.

II. TRADITIONAL APPROACH FOR FAULT LOCALIZATION

Figure 1 shows the traditional fault diagnosis architecture with different modules cooperating to perform fault localization. We later extend this architecture to adapt to the dynamic environment. The topology discovery agents provide the information about the network routes. This information is incorporated into the dependency model. The dependency model contains information about the relationship between the symptoms and the possible faults that can generate these symptoms in the form of dependencies. These dependencies are represented with the probability values based on the degree of confidence in the information. Network monitors probe the network components and observe the network health periodically. They report

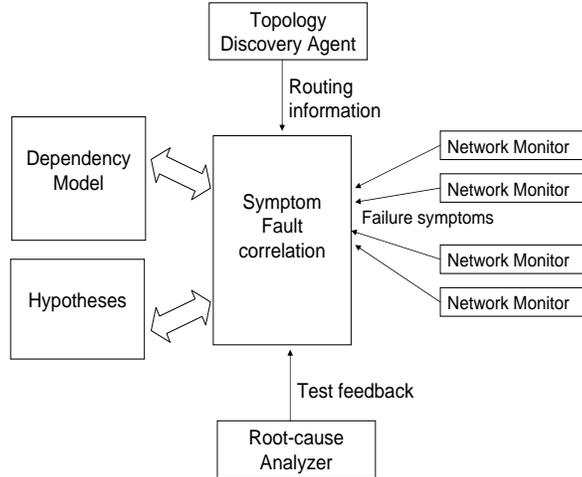


Fig. 1. System architecture for traditional fault localization.

the observed symptoms to the network manager. The manager then performs symptom-fault correlation of these symptoms using the available dependencies, to provide the most probable explanation. The symptom-fault correlation provides a set of hypotheses, each hypothesis consisting of a set of faults that could be possible cause of the observed symptoms. Belief values are computed for these hypotheses indicating the degree of confidence in declaring the hypothesis as the possible cause of failure.

A. Dependency model

Accuracy of fault diagnosis relies upon the accuracy of the dependency model. Since exact causal relationships between faults and symptoms are difficult to determine, dependencies are represented using probabilities, in which the fault nodes are associated with the probability of its independent occurrence of failure and the symptom-fault dependencies are represented with the probability of the causal implication between faults and symptoms.

The dependency model is implemented as a matrix, where rows represent the observable symptoms and the columns represent 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.

B. Symptom fault correlation

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. We use the IHU algorithm [13] for event correlation and belief computation for the hypotheses. Later in this paper we extend this algorithm to incorporate temporal correlation and dynamic dependency

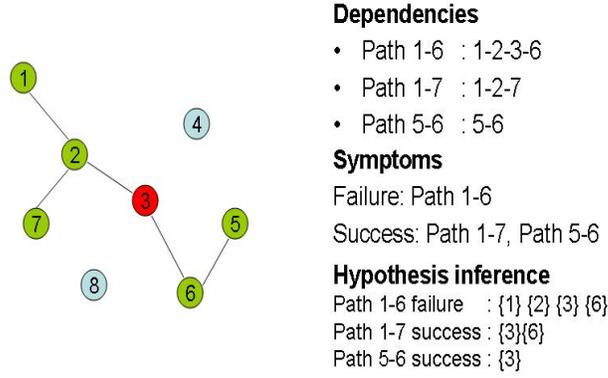


Fig. 2. A network snapshot showing node positions, routes and received symptoms. Figure shows a failure scenario where path 1-6 fails and path 1-7 and path 5-6 succeed; It shows the hypothesis inferred from these symptoms.

changes. The algorithm proceeds in an incremental fashion, with arrival of each new symptom. To incorporate explanations for a symptom S_i , faults explaining S_i are incorporated in the hypothesis. With each new symptom the hypotheses are modified to incorporate explanation for the new symptom. The hypotheses are ranked using a belief metric, which expresses a relative confidence associated with each hypothesis with respect to other hypotheses. In other words, a belief represents relative importance of the hypothesis.

As an example, consider the network shown in Figure 2, which shows the node positions and the routes taken by paths 5-6, 1-7, and 1-6. Consider that a failure is observed on path 1-6, but paths 1-7 and 5-6 show good health. The figure shows a simplistic analysis of the observed failure scenario. Considering only node failures, the figure shows the dependencies between the end-to-end path failures and the node failures. With the available dependency information, failure of path 1-6 can be explained with failure of nodes 1, 2, 3, or 6. However, success of paths 5-6 and 1-7 indicates good health of nodes 1, 2, and 6, leaving node 3 in the suspected node set.

An important task is to compute beliefs for each hypothesis. As described in the IHU algorithm [13], the belief metric contains a negative component and a positive component such that: $b_i(h_j) = \beta b_i^p(h_j) b_i^n(h_j)$, where h_j is the hypothesis, $b_i(h_j)$ is the belief value for h_j , $b_i^p(h_j)$ is the positive belief factor, $b_i^n(h_j)$ is the negative belief factor, and β is the normalizing constant, for the i^{th} iteration.

The negative component represents the belief that the faults forming the hypothesis have occurred and the faults explain all the observed symptoms. The computation is triggered by an observation of the i^{th} symptom, s_i , and a set of hypotheses, H_i , is created, each explaining 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.

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. The positive component is defined as the probability that faults in h_j have not generated any of the observable symptoms.

III. PROPOSED APPROACH FOR ADAPTIVE FAULT LOCALIZATION

Wireless mobile ad-hoc networks possess many unique characteristics [12] because of which existing techniques of fault localization can not be directly used. The problem becomes more challenging in presence of incomplete and inaccurate information due to various factors like lack of static infrastructure, node mobility etc. In this section we discuss our proposed approach to adaptive fault localization.

A. Adaptive approach to build dependency models

Many fault localization algorithms depend on the dependency model [6]. As explained in Section II-A, a dependency model represents the causal relationships between faults and symptoms. Past problem determination work assumes a prior knowledge of symptom-fault dependencies. However in a dynamic environment 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 are continually being added, removed or reconfigured. These dynamic changes affect the symptom-fault dependencies. Figure 3 explains how the change in position of node 3 due to mobility can affect the dependencies and thus in turn affect the inferred hypothesis. This example shows that if the change in routes due to motion of node 3 is not incorporated in the inference mechanism then the same set of reported symptoms can lead to different hypothesis.

As explained in Section II-B, Figure 2 shows the network snapshot at a particular time. It shows the available route information and the hypothesis generated on observing failure of path 1-6 and success of path 1-7 and 5-6. Consider the scenario shown in Figure 3, where node 3 moves causing a change in routes taken by certain end-to-end paths. In this scenario, the same symptoms of failure of path 1-6 and success of paths 1-7 and 5-6 lead to a different hypothesis leaving node 8 in the suspected node set.

- 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 the presence of more than one 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 the following changes:

- 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.

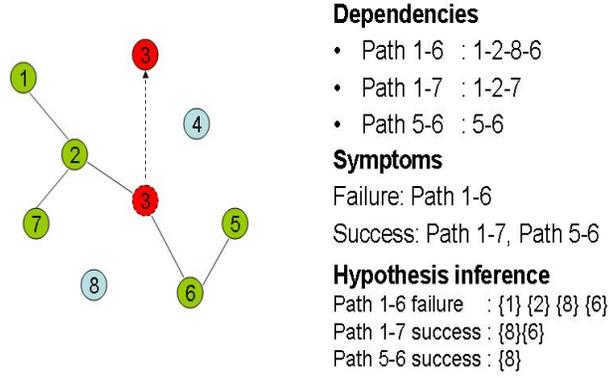


Fig. 3. A snapshot of network shown in Figure 2 at some later time. Here node 3 moves from its position causing a change in the path 1-6, leading to a different hypothesis for the same failure scenario where path 1-6 fails and path 1-7 and path 5-6 succeed.

- Changes in the fault-symptom relationships, which may occur when network topology changes causing existing relationships to become obsolete or new relationships to be added.
- 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.

B. Storing partial fault localization results

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[6] about the 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 that might have caused them. 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 [13]. These are caused by transient network failures or due to overly restrictive threshold values. Including these symptoms in the fault localization can reduce the accuracy of the algorithm. A symptom can be detected to be spurious if it is reported for small time duration. The spurious symptoms can be removed from the HSS if a healthy response for the same probe is obtained within a certain time frame [14]. 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

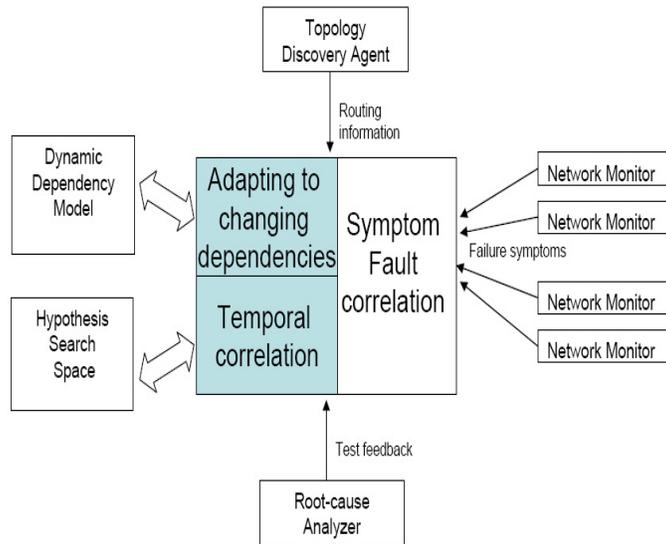


Fig. 4. System architecture for adaptive fault localization.

representing the symptom-fault causal relationship.

C. Incorporating temporal correlation with symptom-fault analysis

As mentioned before, a big challenge is to perform fault diagnosis in parallel with the update in dependency models. The symptom-fault correlation algorithm must be modified to work with symptoms and faults which are added or deleted from the model while a hypothesis construction is ongoing, As a symptom can be explained by different dependency information at different times, it becomes important to incorporate temporal information in the analysis of dependencies. The fact that the time when changed dependencies and symptoms are reported is different from the actual time of occurrence also needs to be incorporated while performing the symptom-fault correlation. We propose a temporal correlation algorithm that extends the existing symptom-fault correlation algorithm by adding a dimension of time in weighing the relevance of dependencies in explaining the observed faults.

IV. SYSTEM ARCHITECTURE FOR ADAPTIVE FAULT LOCALIZATION

Figure 4 extends the traditional architecture shown in Figure 1 to perform adaptive fault localization. Instead of a fixed static dependency model, the presence of mobility demands for a dynamic dependency model, where the same end-to-end path is explained by different failures at different time. These dependencies are reported by topology discovery agents periodically through topology updates. These updates are incorporated into the dynamic dependency model by the adaptive component. The network manager computes hypotheses and stores in the hypotheses search space. The hypotheses search space is updated on receiving dependency updates. Due to the availability of more than one dependency for an end-to-end path, the network manager also performs a temporal correlation while doing this analysis to infer the most relevant cause of failure.

A. Building the dynamic dependency model

The dynamic dependency model is implemented as a set of matrices where each matrix represents dependencies observed at a particular time. As explained in Section II-A, in each matrix rows represent the observable symptoms and the columns represent

the possible network faults. As discussed earlier, the obtained dependency information can be incomplete or inconsistent with the actual view of the network. 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. On receiving an update in the dependency information, the previous information is not completely discarded. Instead, a set of dependency matrices is maintained where dependency information obtained at different points in time is stored in each 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.

B. Building the hypothesis search space

Hypothesis search space (HSS) is a data-structure that explains the symptom-fault dependencies of the observed symptoms. Thus it is also implemented as a matrix representing the symptom-fault relationship, but unlike the dependency model which contains information of all possible symptoms, it stores only the symptoms observed in a time window. On arrival of each symptom reported by the network monitors, the symptom is looked for 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. Maintaining information of each relevant symptom individually also allows updating the probability values of the dependencies 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 observed symptoms.

V. TEMPORAL CORRELATION

The existing Incremental algorithm [13] does not have a notion of time while performing event correlation. In the dynamically changing environment of a 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 symptom-fault relationships: Due to change in the topology the symptom-fault 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. These sets keep getting modified to explain newer symptoms. However 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.

A. *The event correlation process*

In this section we propose an approach to adapt the event correlation approach presented in Section II-B to the changing dependencies. 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 two models with the symptom arrival time. Each dependency model is marked with the timestamp of creation and similarly each symptom is marked with a timestamp of arrival. The inferred symptom-fault dependencies are 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 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 obtaining 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, which also reduces the probability of its being relevant to the reported symptom. This also 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, the row contains the dependency information obtained from the present dependency model. If however a topology update is received within the symptom's time-window, the row contains the 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.

Figure 5 and Figure 6 use the example presented in Figure 2 and Figure 3 to explain the traditional and the adaptive symptom-fault correlation process respectively. These figures show a simplistic view of two dependency models built at a time difference of 100 seconds. The two models show the routes taken by paths 1-6, 1-7, and 5-6. The figures also show the

hypotheses built for a certain failure scenario where path 1-6 failure is reported at 180s and success of path 1-7 and path 5-6 is reported at 190s and 195s respectively.

As shown in Figure 5, in the traditional event correlation, a dependency model is built at time 100s and is used to explain the subsequent symptoms till a new dependency model is created. The dependency model built later at time 200s will then be used to explain the subsequent symptoms occurring after 200s and will not affect the already built hypothesis. Thus in the given failure scenario, the traditional event correlation declares node 3 as the possible cause of failure.

On the other hand, in adaptive event correlation as shown in Figure 6, both the dependency models participate in providing explanation for the symptoms observed in the time window from 100s to 200s, and the explanations from the two models are weighed by the temporal closeness between the symptom arrival time and the model build time. Thus along with the ones and zeros explaining the symptom fault dependencies, the faults are also weighted with the temporal closeness of the symptom arrival time and the dependency model built time.

Figure 6 explains how the same failure scenario shown in Figure 5 is explained differently using adaptive correlation. The matrix shown in the figure represents the hypotheses search space. The following events occur over a time from 100s to 200s: At 100s a dependency model is built. On receiving the failure symptom of path 1-6, an entry is made in the hypotheses search space, indicating that failure of path 1-6 can be explained by failure of nodes 1, 2, 3, or 6. However, this dependency is weighed with the temporal closeness of the model build time (100s) and symptom arrival time (180s). Assuming that dependency discovery is done every 100s, the dependency model built at 100s explains the dependencies at 180s with 20% confidence (while the model built at 200s can explain the same dependencies with 80% confidence). Similarly explanations for the two positive symptoms obtained at time 190s and 195s are weighed based on the temporal closeness to the first dependency model and are entered into the matrix. At 200s, another dependency model is built, and the hypotheses search space is then updated with the explanations obtained with the new dependency model. The newly entered dependencies are also weighed based on the temporal closeness of symptom reporting time and the build time of the new model. For instance, the failure of path 1-6 is explained by the failure of nodes 1, 2, 3, or 8. This dependency model is temporally closer to the symptom arrival time, and thus the explained dependencies for path 1-6 are assigned a higher confidence of 80%.

The hypothesis calculation is done by computing positive and negative beliefs for various failure possibilities. The hypotheses are then ranked based on the belief values and reported. Figure 6 does not show all the belief computation details but presents a simple example to convey the result that could explain the failure using both the dependency models. Failure of path 1-6 is thus explained by nodes 1, 2, 3, 6, and 8 with different confidence values. On the other hand, success of path 1-7 and 5-6 explains good health of nodes 1, 2, 5, 6, and 7 with some confidence, pointing to nodes 3 and 8 with confidence 20% and 80% respectively as possible causes of failure.

The beliefs are computed by the IHU algorithm [13] as mentioned in Section II-B. As we are storing the individual symptom-fault dependencies in the HSS, the 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 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. Using this approach, the positive belief computation involves lesser

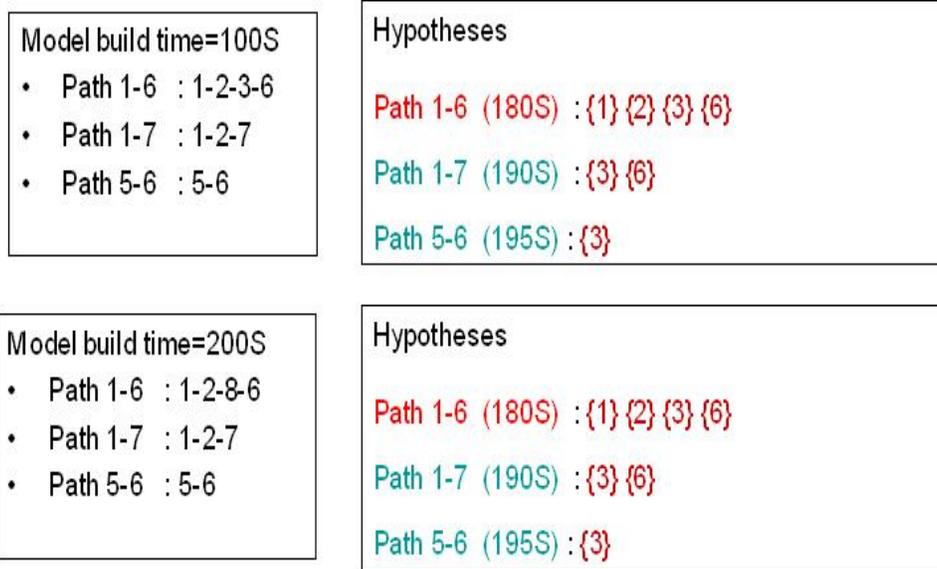


Fig. 5. Example of traditional symptom-fault correlation. Figure shows dependency models built at 100s and 200s and how the model built at 200s is not used in the analysis of a failure scenario where failure of path 1-6 is reported at 180s and success of path 1-7 and 5-6 is reported at 190s and 195s respectively.

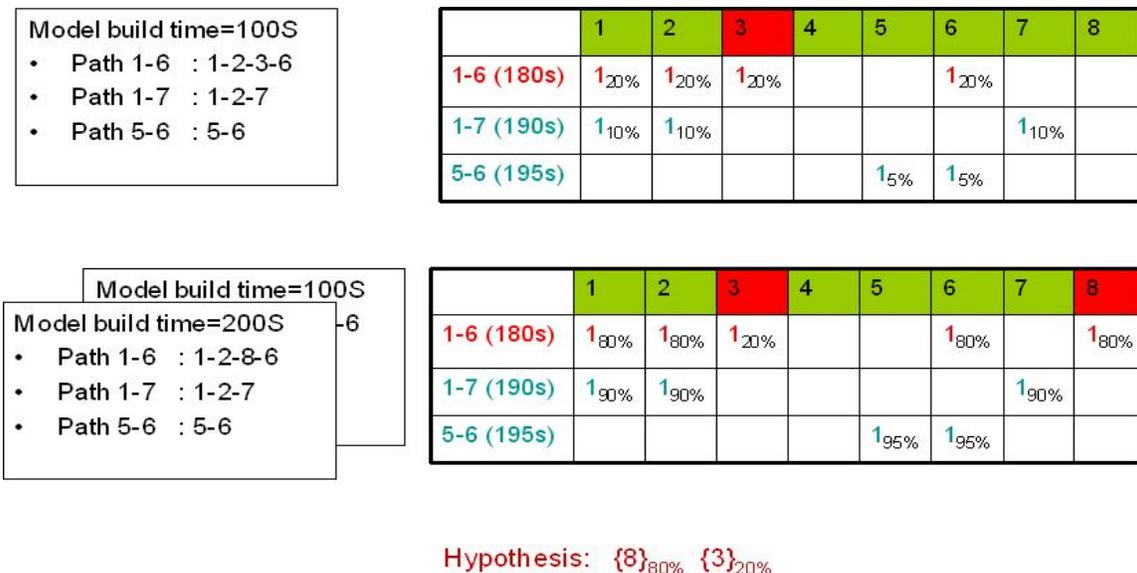


Fig. 6. Example of adaptive symptom-fault correlation. Figure shows dependency models built at 100s and 200s and how both models are used in the analysis of a failure scenario where failure of path 1-6 is reported at 180s and success of path 1-7 and 5-6 is reported at 190s and 195s respectively. Figure shows a hypothesis search space where information from the dependency models is entered to explain the symptoms and is weighed based on the temporal closeness of the symptom arrival time and the model build time.

computation than when done incrementally for each arriving symptom.

VI. 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 a mobile ad-hoc network. We introduce interface failures in the network and then use the proposed algorithm to identify the introduced failure by analyzing the end-to-end failure symptoms. In this problem, the rows of the dependency matrix represent end-to-end connection failures (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. Nodes perform a route discovery to identify the node interfaces that participate in the end-to-end paths. Due to presence of node mobility the end-to-end paths change with time. Hence the nodes periodically perform the topology discovery to obtain information of the changed routes.

Nodes periodically send probes to collect the health status of the hops in various end-to-end paths. Because probes add additional management traffic to the network, it is important to select a set of probes such that all nodes are covered with minimum additional network traffic. Similar issues need to be addressed when sending probes for topology discovery such that the required information is collected with minimum additional traffic and processing overhead. We discuss the probe selection problem in [11].

We observe the effect of degree of mobility and transmission power on the detection rate of the algorithm. With a higher degree of mobility, the end-to-end routes would change more often. Thus the dependency information available in the dependency model might not be accurate, leading to some inaccuracy in the hypotheses computed. Transmission power of a node also affects the detection accuracy. With low transmission power some nodes might not be able to report the failure symptoms to the network manager. Also due to low transmission power managers might not be able to perform a complete topology discovery. Thus low transmission power might lead to low detection rates.

A. Simulation model

We simulated networks of different sizes in Qualnet and introduced interface failures for random time intervals. To observe the effect of mobility and transmission power, we used the random waypoint model for node mobility and observed the accuracy of detection for various values of average node speed and transmission power. We performed experiments on networks with 10, 15, and 20 nodes. We introduced mobility allowing each node to move with average node speed ranging from 3m/s to 11m/s. We varied the transmission power from 11dBm to 20dBm.

We collected the dependency information by periodically running *traceroutes*. Traceroutes provided the information about the nodes involved in various end-to-end paths. We used this information to build the dependency model to represent the dependency of success or failure of end-to-end paths on success or failure of nodes forming the path. We performed topology discovery every 600s by sending traceroutes. To obtain the end-to-end connection failure information we periodically sent *pings* to all the managed nodes in the network. We sent pings on different end-to-end paths of the network with different frequencies ensuring that in every 50s all nodes are probed at least once by some ping. We sent a train of pings on the selected paths and declared an end-to-end path failure on observing an average loss rate higher than 80%. Thus a ping success or failure provides a positive or negative symptom respectively for the event correlation process.

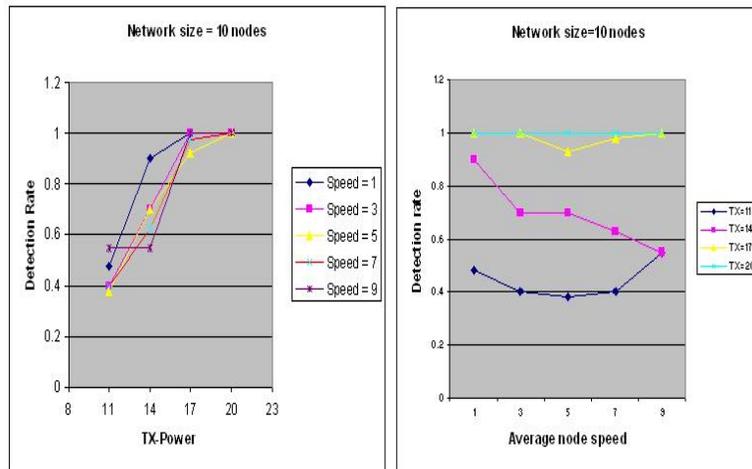


Fig. 7. Effect of transmission power and node speed on detection rate for network with 10 nodes

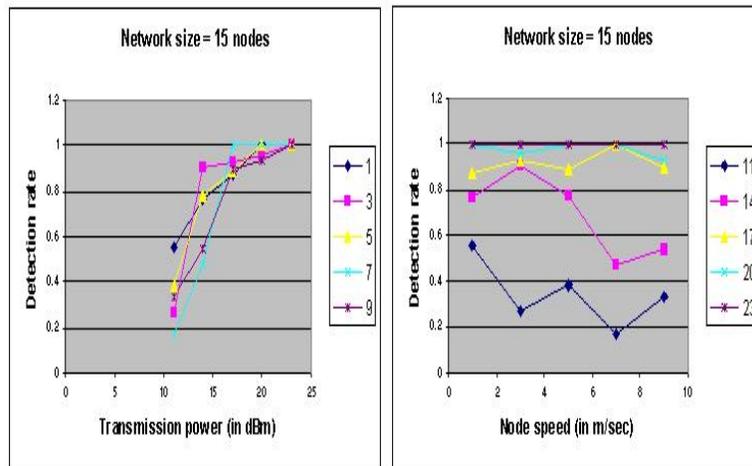


Fig. 8. Effect of transmission power and node speed on detection rate for network with 15 nodes

We computed the detection rate of the algorithm to identify the closeness in the fault declared by the algorithm and the actual fault. The algorithm generates a set of hypothesis and computes belief values indicating the algorithm's confidence in the hypothesis. The hypothesis with highest belief indicates the most probable explanation computed by algorithm for the observed scenario. The algorithm shows perfect accuracy when the actual fault is declared in the hypothesis with highest belief and it behaves worst when the actual fault is not present in any of the computed hypotheses sets. Hence we classified the experient results into three classes, and assigned values between 0 and 1 to these classes to indicate the accuracy of the declared hypotheses. An experiment result belongs to class 1 if the actual fault is present in the hypothesis with highest belief. It belongs to class 2 if the actual fault is present in the hypotheses with lower 3 belief values. It belongs to class 3 otherwise. An experiment with result in class 1, 2, and 3 are assigned values 1, 0.5, and 0 respectively. Each point on the graph represents an average of 40 runs.

Figures 7, 8, and 9 show the effect of average node speed and transmission power on the detection rate for a network with

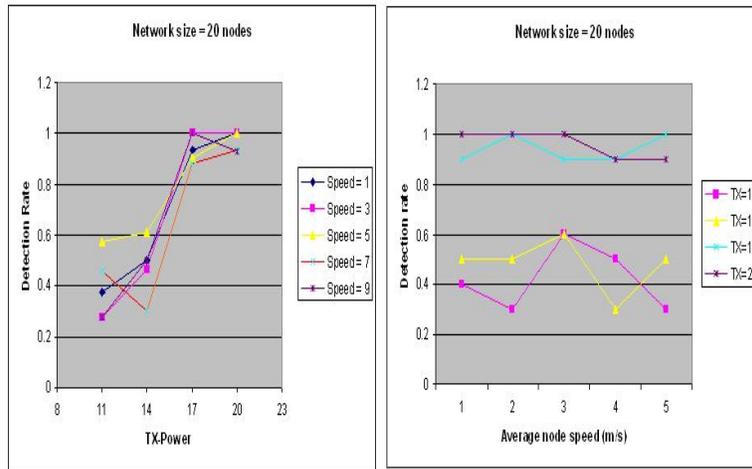


Fig. 9. Effect of transmission power and node speed on detection rate for network with 20 nodes

10, 15, and 20 nodes respectively. We vary average node speeds from 3m/s to 11m/s, and transmission power from 11dBm to 20dBm.

We observed that for higher speeds the detection rate decreases. For various values of transmission power, the detection rate tends to decrease with increase in speed. The decrease in detection rate with increased mobility can be explained by the lack of accuracy in the dependency information inferred about the network. With increase in node's speed, the node position changes faster causing changes in routes, thereby causing more frequent changes in the network topology. These changes are not accurately retrieved by the *traceroutes*, causing an inaccuracy in the dependency model. Thus the reported path failure is not explained by the actual path dependencies at the time of failure. This analysis suggests a need to adapt the frequency of topology discovery process to the degree of mobility present in the system. If the mobility is high, the topology discovery should be more frequent.

We observed that with increase in transmission power the detection rate improves. For low transmission power, low detection rate can be explained by the inability of the node to report symptoms to the network manager. Thus the fault diagnosis is done with incomplete positive and negative symptoms. Also low transmission power leads to the inability of the network manager to extract complete routing information resulting in an incomplete dependency model. Thus incomplete information is available to the network manager about the reported path failures leading to less accurate fault diagnosis. Extremely low power levels can also lead to transient pockets of disconnected nodes.

We also observed that detection rate reaches higher values beyond a threshold. With smaller speeds this threshold is reached for smaller transmission power, while for higher speeds it takes higher transmission power to reach such a threshold. For instance, in Figure 8 for a speed of 3m/s, the threshold is reached for transmission power of 14dBm while for speed of 11m/s, it is reached at transmission power of 17 dBm.

VII. CONCLUSION AND FUTURE WORK

In this paper, we described a strategy for fault diagnosis in a dynamically changing environment of mobile ad-hoc networks. We designed a system architecture that adapts to changing dependencies in the network. The architecture uses a dynamic

dependency model and hypotheses search space to incorporate the observed changes. We illustrated a method to perform temporal correlation and proposed an adaptive fault diagnosis algorithm. The algorithm computes a set of hypotheses as possible causes of failure and ranks them with the degree of confidence. We evaluated the proposed method by simulations in Qualnet and presented the effect of mobility and transmission power on the detection rate of the fault diagnosis algorithm.

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] K. Appleby, G. Goldszmidt, and M. Steinder. Yemanja - a layered event correlation system for multi-domain computing utilities. *Journal on Network and Systems Management*, 10(2):171–194, Jun., 2002.
- [2] B. Gruschke. Integrated event management, event correlation using dependency graphs. In *Distributed Systems Operations Management*, pages 130–141, 1998.
- [3] G. Jakobson and M. D. Weissman. Alarm correlation. *IEEE Network*, pages 52–59, Nov., 1993.
- [4] 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.
- [5] L. Kant, A. S. Sethi, and M. Steinder. Fault localization and self-healing mechanisms for fcs networks. In *23rd Army Science Conference, Orlando, FL*, Dec. 2002.
- [6] I. Katzela and M. Schwartz. Schemes for fault identification in communication networks. *IEEE/ACM Transactions on Networking*, 3(6):733–764, 1995.
- [7] S. Kliger, S. Yemini, Y. Yemini, D. Ohsie, and S. Stolfo. A coding approach to event correlation. In *Fourth International Symposium on Intelligent Network Management*, 1995.
- [8] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless network research. Technical report, Dartmouth College.
- [9] G. Liu, A. K. Mok, and E. J. Yang. Composite events for network event correlation. *Integrated Network Management, VI*, pages 247–260, May., 1999.
- [10] M. Natu and A. S. Sethi. Adaptive fault localization in mobile ad-hoc battlefield networks. In *MILCOM'05, Atlantic City, NJ*, 2005.
- [11] M. Natu and A. S. Sethi. Active probing approach for fault localization in computer networks. In *E2EMON'06, Vancouver, Canada*, 2006.
- [12] L. Qiu, P. Bahl, A. Rao, and L. Zhou. Fault detection, isolation, and diagnosis in multihop wireless networks. Technical report, Microsoft Research, Redmond, WA, 2003.
- [13] M. Steinder and A. S. Sethi. Probabilistic fault diagnosis in communication systems through incremental hypothesis updating. *Computer Networks*, 45(4):537–562, July 2004.
- [14] S. A. Yemini, S. Kliger, E. Mozes, Y. Yemini, and D. Ohsie. High speed and robust event correlation. *IEEE Communications Magazine*, 34(5):82–90, 1996.