Active Integrated Fault Localization in Communication Networks

Yongning Tang and Ehab S. Al-Shaer
School of Computer Science
DePaul University
243 S. Wabash
Chicago, IL, USA
{ytang, ehab}@cs.depaul.edu

Raouf Boutaba
School of Computer Science
University of Waterloo
200 University Ave. W.
Waterloo, ON, Canada
rboutaba@bbcr.uwaterloo.ca

Abstract

Fault localization is a core element in fault management. Many fault reasoning techniques use deterministic or probabilistic symptom-fault causality model for fault diagnoses and localization. Symptom-Fault map is commonly used to describe Symptom-Fault causality in fault reasoning. However, due to lost and spurious symptoms in fault reasoning systems that passively collect symptoms, the performance and accuracy of the fault localization can be significantly degraded. In this paper, we propose an extended Symptom-Fault-Action model to incorporate actions into fault reasoning process to tackle the above problem. This technique is called Active Integrated fault Reasoning (*AIR*), which contains three modules: fault reasoning, fidelity evaluation and action selection. Corresponding fault reasoning and action selection algorithms are elaborated. Simulation study shows both performance and accuracy of fault reasoning can be greatly improved by taking actions, especially when the rate of spurious and lost symptoms is high.

Keywords

fault management, fault reasoning, action selection, probabilistic inference

1. INTRODUCTION

Fault localization is a basic component in fault management system because it identifies the fault reason which can best explain the observed network disorders. Most fault reasoning algorithms use a bipartite directed acylic graph to describe the Symptom-Fault correlation, which represents the causal relationship between each fault f_i and a set of its observed symptoms S_{f_i} [4]. Symptom-Fault causality graph provides a vector of correlation likelihood measure $p(s_i|f_i)$, to bind a fault f_i to a set of its symptoms S_{f_i}.

Two approaches are commonly used in fault reasoning and localization: passive diagnosis ([2], [4], [3], [7] and active probing ([6], [5], [1], [9]). In passive approach, all symptoms are passively collected and then processed to infer the root faults. In active approach, faults are detected by conducting a set of probing actions. Passive approach causes less intrusiveness in management networks. However, it may take long time to discover the root faults, particularly if symptom loss ratio is high. On the other hand, although active probing approach is more efficient to identify faults quickly, probing might cause significant overhead particularly in large-scale networks. In this paper, we propose

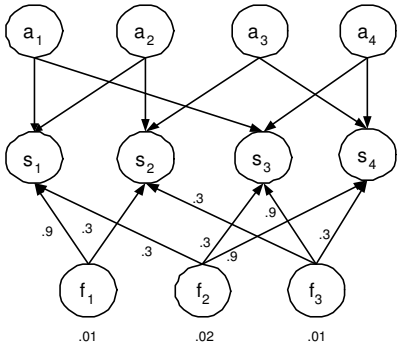

Figure 1: Action-Symptom-Fault Model

a novel fault localization technique that integrates the advantage of both passive and active monitoring into one framework, called *Active Integrated fault Reasoning* or *AIR*. In our approach, if the passive reasoning is not sufficient to explain the problem, AIR selects optimal probing actions to discover the most critical symptoms that are important to explain the problem but they have been lost or corrupted during passive fault reasoning. Our approach significantly improves the performance of fault localization while minimizing the intrusiveness of active fault reasoning.

AIR consists of three modules: Fault Reasoning (FR); Fidelity Evaluation (FE); and Action Selection(AS). *Fault reasoning* module passively analyzes observed symptoms and generates a fault hypothesis. The fault hypothesis is then sent to *fidelity evaluation* module to verify if the fidelity value of the reasoning result is satisfactory. If the correlated symptoms necessary to explain the fault hypothesis are observed (i.e. high fidelity), then fault reasoning process terminates. Otherwise, a list of most likely unobserved symptoms that can contribute to the fault hypothesis fidelity is sent to the *action selection* module, which then performs selected actions to determine which symptoms has occurred but not observed (i.e. lost) and accordingly adjust hypothesis fidelity value. If the new fidelity value is satisfactory, then the reasoning process terminates; otherwise, the new symptom evidence is fed into the *fault reasoning* module to create a new hypothesis. This process is recursively invoked until a highly credible hypothesis is found.

The paper is organized as follows. In section 2, we discuss our research motivation and the problem formalization. In section 3, we describe the components and algorithms of AIR. In Section 4, we present a simulation study to evaluate AIR performance and accuracy. In Section 5, related work is discussed. In section 6, we give our conclusion and future work.

2. MOTIVATION AND PROBLEM FORMALIZATION

In general, active fault management does not scale well when number of managed nodes or faults grow significantly in the network. In fact, some faults such as intermittent reachability problem may not even be identified if only active fault management is used. However, this can be easily reported using passive fault management systems because agents are configured to report abnormal system conditions or symptoms such as high average

Notation	Definition
S_{f_i}	a set of all symptoms caused by the fault f_i
F_{s_i}	a set of all faults that might cause symptom s_i
S_O	a set of all observed symptoms so far
S_{O_i}	a set of *observed* symptoms caused by fault f_i
S_{U_i}	a set of *not-yet-observed* (lost) symptoms caused by the fault f_i
h_i	a set of faults that constitute a possible hypothesis that can explain S_O
Φ	a set of all different fault hypotheses, h_i, that can explain S_O
S_N	a set of correlated but not-yet-observed symptoms associated with any fault in a hypothesis
S_V	a subset of S_N, which includes symptoms that their *existence* is confirmed
S_U	a subset of S_N, which includes symptoms that their *non-existence* is confirmed

Figure 2: Active Integrated Fault Reasoning Notation

packet drop ratio. On the other hand, symptoms can be lost due to noisy or unreliable communications channels, or they might be corrupted due to spurious (untrue) symptoms generated as a result of malfunctioning agents or devices. This significantly reduces the accuracy and the performance of passive fault localization. Only the integration of active and passive reasoning can provide efficient fault localization solutions.

To incorporate actions into traditional Symptom-Fault model, we propose an extended Symptom-Fault-Action model as shown in Fig. 1. In our model, actions are properly selected probes or test transactions that are used to detect or verify the existence of observable symptoms. Actions can simply include commonly used network utilities, like ping and traceroute; or some proprietary fault management system, like SMRM [1]. We assume that symptoms are verifiable, which means that, if the symptom ever occurred, we could verify the symptom existence by executing some probing actions or checking the system status such as system logs.

In this paper, we use $F = \{f_1, f_2, \ldots, f_n\}$ to denote the *fault set*, and $S = \{s_1, s_2, \ldots, s_m\}$ to denote the *symptom set* that can be caused by one or multiple faults in F. Causality matrix $P_{F\times S} = \{p(s_i|f_j)\}$ is used to define causal certainty between fault $f_i (f_i \in F)$ and symptom $s_i (s_i \in S)$. If $p(s_i|f_j) = 0$ or 1 for all (i, j), we call such causality model a deterministic model; otherwise, we call it a probabilistic model. We also use $A = \{a_1, \ldots, a_k\}$ to denote the list of actions that can be used to verify symptom existence. We describe the relation between actions and symptoms using *Action Codebook* represented as a bipartite graph as shown in Fig. 1. For example, the symptom s_1 can be verified using action a_1 or a_2. The Action Codebook can be defined by network managers based on symptom type, the network topology, and the available fault diagnostic tools. The extended Symptom-Fault-Action graph is viewed as a 5-tuple (S, F, A, E_1, E_2), where fault set F, symptom set S, and action set A are three independent vertex sets. Every edge in E_1 connects a vertex in S and another vertex in F to indicate causality relationship between symptoms and faults. Every edge in E_2 connects a vertex in A and another vertex in S to indicate the Action Codebook. For convenience, in Fig. 2, we introduce the notations used in our discussion throughout this paper. The basic Symptom-Fault-Action model can be described as the following:

- For every action, associate an action vertex a_i, $a_i \in A$;
- For every symptom, associate a symptom vertex s_i, $s_i \in S$;

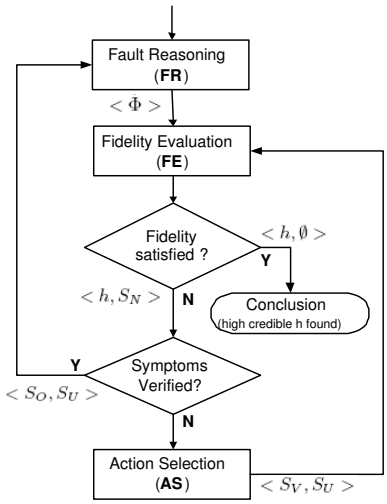

Figure 3: Active Action Integrated Fault Reasoning

- For every fault, associate a fault vertex f_i, $f_i \in F$;
- For every fault f_i, associate an edge to each s_i caused by this fault with a weight equal to $p(s_i|f_i)$;
- For every action a_i, associate an edge of weight equal to the action cost to each symptom verifiable by this action.

The performance and accuracy are the two most important factors for evaluating fault localization techniques. Performance is measured by fault detection time T, which is the time between receiving the fault symptoms and identifying the root faults. The fault diagnostic accuracy depends on two factors: (1) the detection ratio (α), which is the ratio of the number of *true* detected root faults (F_d is the total detected fault set) to the number of *actual* occurred faults F_h, formally $\alpha = \frac{|F_d \cap F_h|}{|F_h|}$; and (2) false positive ratio (β), which is the ratio of the number of *false* reported faults to the total number of detected faults; formally $\beta = \frac{|F_d - F_d \cap F_h|}{|F_d|}$ [4]. Therefore, the goal of any fault management system is to increase α and reduce β in order to achieve high accurate fault reasoning results.

The task of the fault reasoning is to search for root faults in F based on the observed symptoms S_O. Our objective is to improve fault reasoning by minimizing the detection time, T and the false positive ratio, β, and maximizing the detection ratio, α.

In order to develop this system, we have to address the following three problems: (1) Given the Fault-Symptom correlation matrix and the set of observed symptoms (S_O), construct a set of the most possible hypotheses, $\Phi = \{h_1, h_2, \ldots, h_p\}, h_i \subseteq F$, that can explain the current observed symptoms; (2) Given a set of possible hypotheses, find the most credible hypothesis h, that can give the best explanation for the current observed symptoms; (3) If the selected hypothesis does not satisfy fidelity requirement, then given the unobserved symptoms S_N and select the minimum-cost actions to search for an acceptable hypothesis. In the following, we will discuss the solution for each problem.

3. ACTIVE INTEGRATED FAULT REASONING

The Active Integrated Fault Reasoning (AIR) process (Fig. 3) includes three functional modules: Fault Reasoning (FR), Fidelity Evaluation (FE), and Action Selection (AS). The Fault Reasoning module takes passively observed symptoms S_O as input and returns fault hypothesis set Φ as output. The fault hypothesis set Φ might include a set of hypotheses ($h_1, h_2, \ldots, h_n$) where each one contains a set of faults that explains all observed symptoms so far. Then, Φ is sent to the Fidelity Evaluation module to check if any hypothesis hi ($h_i \in \Phi$) is satisfactory. If most correlated symptoms necessary to explain the fault hypothesis h_i are observed (i.e. high fidelity), then the Fault Reasoning process terminates. Otherwise, a list of unobserved symptoms, S_N, that contribute to explain the fault hypothesis h_i of the highest fidelity, is sent to the Action Selection module to determine which symptoms have occurred. As a result, the fidelity value of hypothesis h_i is adjusted accordingly. The conducted actions return the test result with a set of existing symptoms S_V and non-existing symptoms S_U. The corresponding fidelity value might be increased or decreased based on the action return results. If the newly calculated fidelity is satisfied, then the reasoning process terminates; otherwise, S_O, S_U are sent as new input to the Fault Reasoning module to create a new hypothesis. This process is repeated until a hypothesis with high fidelity is found. Fidelity calculation is explained later in this section. In the following, we describe the three modules in detail, then discuss the complete Active Integrated Fault Reasoning algorithm.

3.1 Heuristic Algorithm for Fault Reasoning

In the Fault Reasoning module, we use a *contribution function*, $C(f_i)$, as a criteria to find faults that have the maximal contribution of the observed symptoms. In the probabilistic model, symptom s_i can be caused by a set of faults $f_i, (f_i \in F_{s_i})$ with different possibilities $p(s_i|f_i) \in (0, 1]$. We assume that the Symptom-Fault correlation model is sufficient enough to neglect other undocumented faults (i.e., prior fault probability is very low). Thus, we can also assume that symptom s_i will not occur if none of the faults in F_{s_i} happened. In other words, if s_i occurred, at least one f_i ($f_i \in F_{s_i}$) must have occurred. However conditional probability $p(s_i|f_i)$ itself may not truly reflect the chance of fault f_i occurrence by observing symptom s_i. For example, in Fig. 1, by observing s_1, there are three possible scenarios: f_1 happened, f_2 happened or both happened. Based on the heuristic assumption that the possibility of multiple faults happened simultaneously is low, one of the faults (f_1 or f_2) should explain the occurrence of s_1. In order to measure the contribution of each fault f_i to the creation of s_i, we normalize the conditional probability $p(s_i|f_i)$ to the normalized conditional probability $\hat{p}(s_i|f_i)$ to reflect the relative contribution of each fault f_i to the observation of s_i.

$$\hat{p}(s_i|f_i) = \frac{p(s_i|f_i)}{\sum_{f_i \in F_{s_i}} p(s_i|f_i)} \tag{1}$$

With $\hat{p}(s_i|f_i)$, we can compute normalized posterior probability $\hat{p}(f_i|s_i)$ as follows.

$$\hat{p}(f_i|s_i) = \frac{\hat{p}(s_i|f_i)p(f_i)}{\sum_{f_i \in F_{s_i}} \hat{p}(s_i|f_i)p(f_i)} \tag{2}$$

$\hat{p}(f_i|s_i)$ shows the relative probability of f_i happening by observing s_i. For example, in Fig. 1, assuming all faults have the same prior probability, then $\hat{p}(f_1|s_1) = 0.9/(0.9 + 0.3) = 0.75$ and $\hat{p}(f_2|s_1) = 0.3/(0.9 + 0.3) = 0.25$. The following contribution function $C(f_i)$ evaluates all contribution factors $\hat{p}(f_i|s_i)$, $s_i \in S_{O_i}$ with the observation S_{O_i}, and decides which f_i is the best candidate with maximum contribution value $C(f_i)$ to the currently not yet explained symptoms.

$$C(f_i) = \frac{\sum_{s_i \in S_{O_i}} \hat{p}(f_i|s_i)}{\sum_{s_i \in S_{f_i}} \hat{p}(f_i|s_i)} \tag{3}$$

Therefore, fault reasoning becomes a process of searching for the fault (f_i) with maximum $C(f_i)$. This process continues until all observed symptoms are explained. The contribution function $C(f_i)$ can be used for both deterministic and probabilistic model.

In the deterministic model, the more the number of symptoms observed, the stronger the indication that the corresponding fault has occurred. Meanwhile, we should not ignore the influence of prior fault probability $p(f_i)$, which represents long-term statistical observation. Since $p(s_i|f_j) = 0$ or 1 in the deterministic model, the normalized conditional probability reflects the influence of prior probability of fault f_i. Thus, the same contribution function can seamlessly combine the effect of $p(f_i)$ and the ratio of $\frac{|S_{O_i}|}{|S_{f_i}|}$ together.

In the fault reasoning algorithm, first it finds the fault candidate set F_C including all faults that can explain at least one symptom s_i ($s_i \in S_O$), then it calls the function $HU()$ to generate and update the hypothesis set Φ until all observed symptoms S_O can be explained. According to the contribution $C(f_i)$ of each fault f_i ($f_i \in F_C$), algorithm 1 searches for the best explanation of S_K, which is currently observed but not yet explained symptom by the hypothesis h_i (lines 2-12). Here $S_K = S_O - \cup_{f_i \in h_i} S_{O_i}$ and initially $S_K = S_O$. If multiple faults have same contribution, multiple hypotheses will be generated (lines 13-17). The searching process (HU) will recursively run until all observed symptoms explained (lines 18-24). Notice that only those hypotheses with minimum number of faults that cover all observed symptoms are included into Φ (lines 23-24).

The above Fault Reasoning algorithm can be applied to both deterministic and probabilistic models with same contribution function $C(f_i)$ but different conditional probability $p(s_i|f_i)$.

3.2 Fidelity Evaluation of Fault Hypotheses

The fault hypotheses created by the Fault Reasoning algorithm may not accurately determine the root faults because of lost or spurious symptoms. The task of the Fidelity Evaluation is to measure the credibility of hypothesis created in the reasoning phase given the corresponding observed symptoms. How to objectively evaluate the reasoning result is crucial in fault localization systems.

We use the fidelity function $FD(h)$ to measure the credibility of hypothesis h given the symptom observation S_O. We assume that the occurrence of each fault is independent.

Algorithm 1 Hypothesis Updating Algorithm **HU(**h, S_K, F_P)

Input: hypothesis h, observed but uncovered symptom set S_K, fault candidate set F_P
Output: fault hypothesis set Φ

```
1: c_max = 0
2: for all f_i ∈ F_P do
3:    if C(f_i) > c_max then
4:       c_max ← C(f_i)
5:       F_S ← ∅
6:       F_S ← F_S ∪ {f_i}
7:    else
8:       if C(f_i) = c_max then
9:          F_S ← F_S ∪ {f_i}
10:      end if
11:   end if
12: end for
13: for all f_i ∈ F_S do
14:   h_i ← h ∪ {f_i}
15:   S_{K_i} ← S_K − S_{O_i}
16:   F_{P_i} ← F_P − {f_i}
17: end for
18: for all S_{K_i} = ∅ do
19:   if S_{K_i} = ∅ then
20:      Φ ← Φ ∪ {h_i}
21:   end if
22: end for
23: if Φ ≠ ∅ then
24:   return < Φ >
25: else
26:   /* No h_i can explain all S_O*/
27:   for all h_i do
28:      HU(h_i, S_{K_i}, F_{P_i})
29:   end for
30: end if
```

- For deterministic model:

$$FD(h) = \frac{\sum_{f_i \in h} |S_{O_i}|/|S_{f_i}|}{|h|} \tag{4}$$

- For probabilistic model:

$$FD(h) = \frac{\prod_{s_i \in \bigcup_{f_i \in h} S_{f_i}} (1 - \prod_{f_i \in h} (1 - p(s_i|f_i)))}{\prod_{s_i \in S_O} (1 - \prod_{f_i \in h} (1 - p(s_i|f_i)))} \tag{5}$$

Obviously in the deterministic model, if the hypothesis h is correct, $FD(h)$ must be equal to 1 because the corresponding symptoms can be either observed or verified. In the probabilistic model, if related symptoms are observed or verified, $FD(h)$ of a credible hypothesis can still be less than 1 because some symptoms may not happen even when the hypotheses are correct. In either case, our fidelity algorithm takes in consideration a target Fidelity Threshold, $FD_{THRESHOLD}$, that the user can configure to accept hypothesis. System administrators can define the threshold based on long-term observation and previous experience. If the threshold is set too high, even correct hypothesis will be ignored; but if the threshold is too low, then less credible hypothesis might be selected.

Fidelity evaluation function is used to evaluate each hypothesis and decides if the result is satisfactory by comparing to the pre-defined threshold value. If an acceptable hypothesis that matches the fidelity threshold exists, the fault localization process can

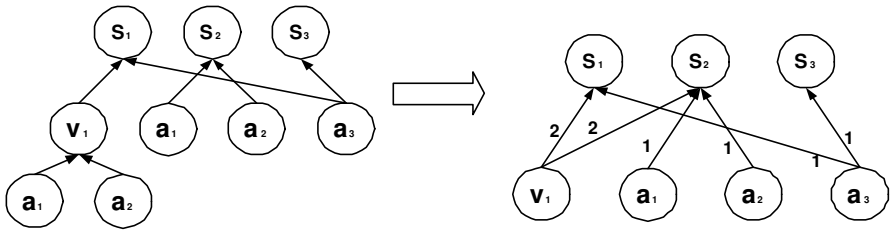

Figure 4: Symptom-Action Bipartite Graph

terminate. Otherwise, the best available hypothesis and a non-empty set of symptoms (S_N) would be verified in order to reach a satisfactory hypothesis in the next iteration.

3.3 Action Selection Heuristic Algorithm

The main reason to verify the existence of symptoms rather than faults is that symptoms are noticable/visible consequences of faults and thus they are easier to track and verify. The task of Action Selection is to find the least-cost actions to verify S_N (unobserved symptoms) of the hypothesis that has highest fidelity. As the size of S_N grows very large, the process of selecting the minimal cost action that verifies S_N becomes non-trivial. The Action-Symptoms correlation graph can be represented as a 3-tuple (A, S, E) graph such that A and S are two independent vertex sets representing Actions and Symptoms respectively, and every edge e in E connects a vertex $a_j \in A$ with a vertex $s_i \in S$ with a corresponding weight (w_{ij}) to denote that a_j can verify s_i with cost $w_{ij} = w(s_i, a_j) > 0$. If there is no association between s_i and a_j, then w_{ij} = 0. Because a set of actions might be required to verify one symptom, we use a virtual action vertex, v_j, to represent this case. The virtual action vertex v_j is used to associate a set of conjunctive actions to the corresponding symptom(s). However, if multiple actions are directly connected to a symptom, then this means any of these actions can be used disjunctively to verify this symptom (Fig. 4). To convert this to a bipartite graph, (1) we set the weight of v_j, $w(s_i, v_j)$, to the total cost of the conjunctive action set, (2) then eliminate the associated conjunctive set to the v_j, (3) associate v_j with all symptoms that can be verified by any action in the conjunctive action set.

The Symptom-Action graph in Fig. 4 presents the verification relationship between symptoms $\{s_1, s_2, s_3\}$ and actions $\{a_1, a_2, a_3\}$. Symptom s_1 can be verified by taking a combination of action a_1 and a_2, which causes a new virtual action vertex v_1 to be created with weight 2. Action v_1 can verify all symptoms (s_1, s_2) that are verifiable by either a_1 or a_2. After converting action combination to a virtual action, Symptom-Action correlation can be represented in a bipartite graph.

The goal of the Action Selection algorithm is to select the actions that cover all symptoms S_N with a minimal action cost. With the representation of Symptom-Action bipartite graph, we can model this problem as a weighted set-covering problem. Thus, the Action Selection algorithm searches for A_i such that A_i includes the set of actions that cover all the symptoms in the Symptoms-Action correlation graph with total minimum cost. We can formally define A_i as the covering set that satisfies the following conditions: (1) $\forall s_i \in S$, $\exists a_j \in A_i$ s.t. $w_{ij} > 0$, and (2) $\sum_{a_i \in A_i, s_j \in S_N} w_{ij}$ is the *minimum*.

Algorithm 2 Active Integrated Fault Reasoning S_O

Input: S_O
Output: fault hypothesis h

$S_N \leftarrow S_O$
while $S_N \neq \emptyset$ **do**
  $\Phi = FR(S_O)$
  $< h, S_N >= FE(\Phi)$
  if $S_N = \emptyset$ **then**
    return $< h >$
  else
    if IPP experied **then**
      /*used to schedule active fault localization periodically*/
      $< S_V, S_U >= AS(S_N)$
    end if
  end if
  $S_O \leftarrow S_O \cup S_V$
  $< h, S_N >= FE(\{h\})$
  if $S_N = \emptyset \parallel S_V = \emptyset$ **then**
    return $< h >$
  end if
end while

The weighted set-covering is an NP-complete problem. Thus, we developed a heuristic greedy set-covering approximation algorithm to solve this problem. The main idea of the Algorithm is simply first selecting the action (a_i or v_i) that has the maximum *relative covering ratio*, $R_i = \frac{|S_{a_i}|}{\sum_{s_j \in S_{a_i}} w_{ij}}$, where this action is added to the final set A_f and removed from the candidate set A_c that includes all actions. Here, S_{a_i} is the set of symptoms that action a_i can verify, $S_{a_i} \subseteq S_N$. Then, we remove all symptoms that are covered by this selected action from the unobserved symptom set S_N. This search continues to find the next action a_i ($a_i \in A_c$), that has the maximum ratio R_i until all symptoms are covered (i.e., S_N is empty). Thus, intuitively, this algorithm appreciates actions that have more symptom correlation or aggregation. If multiple actions have the same relative covering weight, the action with more covered symptoms (i.e., larger $|S_{a_i}|$ size) will be selected. If multiple actions have the same ratio, R_i, and same $|S_{a_i}|$, then each action is considered independently to compute the final selected sets for each action and the set that has the minimum cost is selected. Finally, it is important to notice that each single action in the A_f set is necessary for the fault determination process because each one covers unique symptoms.

3.4 Algorithm for Active Integrated Fault Reasoning

The major contribution of this work is to incorporate active actions into fault reasoning. Passive fault reasoning could work well if enough symptoms can be observed correctly. However in most cases, we need deal with interference from symptom loss and spurious symptoms, which could mislead fault localization analysis. As a result of fault reasoning, the generated hypothesis suggests a set of selected symptoms S_N that are unobserved but expected to happen based on the highest fidelity hypothesis. If fidelity evaluation of such hypothesis is not acceptable, optimal actions are selected to verify S_N. Action results will either increase fidelity evaluation of previous hypothesis or bring new evidence to generate new hypothesis. By taking actions selectively, the system can evaluate fault hypotheses progressively and reach to root faults.

(a) (b) (c)

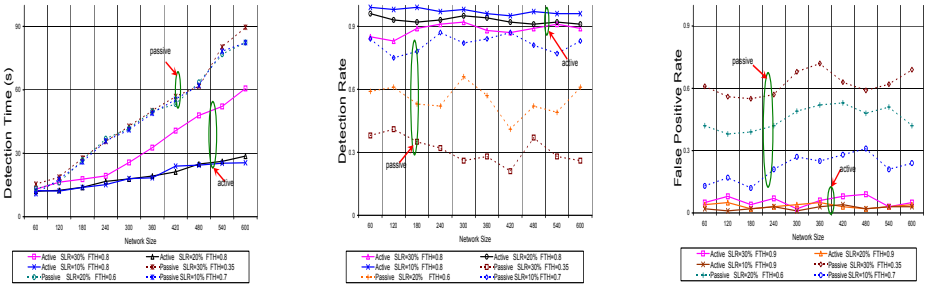

Figure 5: The Impact of Symptom Loss Ratio (a) Detection Time T (b) Detection rate α (c) False positive rate β

Algorithm 2 illustrates the complete process of the AIR technique. Initially, the system takes observed symptom S_O as input. Fault Reasoning is used to search the best hypothesis Φ (Line 3). Fidelity is the key to associate passive reasoning to active probing. Fidelity Evaluation is used to measure the correctness of corresponding hypothesis h ($h \in \Phi$), and produce expected missing symptoms S_N (Line 3). If the result h is satisfied, the process terminates with current hypothesis as output (Line 5 - 6). Otherwise, AIR waits until Initial Passive Period (IPP) expired (Line 8) to initiate actions to collect more evidence of verified symptoms S_V and not-occurred symptoms S_U (Line 10). New evidence will be added to re-evaluate previous hypothesis (Line 13). If fidelity evaluation is still not satisfied, the new evidence with previous observation is used to search another hypothesis (Line 3) until the fidelity evaluation is satisfied. At any point, the program terminates and returns the current selected hypothesis, if either the fidelity evaluation does not find symptoms to verify (S_N is $\emptyset$), or none of the verified symptom had occurred (S_V is $\emptyset$). In either case, this is an indication that the current selected hypothesis is creditable.

4. SIMULATION STUDY

In this section, we describe our simulation study to evaluate Action Integrated fault Reasoning (AIR) technique. We conducted a series of experiments to measure how AIR improves the performance and the accuracy of the fault localization compared with Passive Fault Reasoning (PFR). The evaluation study considers fault detection time T as a performance parameter and the detection rate α and false positive rate β as accuracy parameters.

In our simulation study, the number of monitored network objects D ranged from 60 to 600. We assume every network object can generate different faults and each fault could be associated with 2 to 5 symptoms uniformly distributed. The number of simulated symptoms vary from 120 to 3000 uniformly distributed. We use fault cardinality (FC), symptom cardinality (SC) and action cardinality (AC) to describe the Symptom-Fault-Action matrix such that FC defines the maximal number of symptoms that can be associated with one specific fault; SC defines the maximal number of faults one symptom might correlate to; AC defines the maximal number of symptoms that one action can verify. The independent prior fault probabilities $p(f_i)$ and conditional probabilities $p(s_i|f_j)$ are uniformly distributed in ranges [0.001, 0.01] and (0, 1] respectively. Our simulation model

(a) (b) (c)

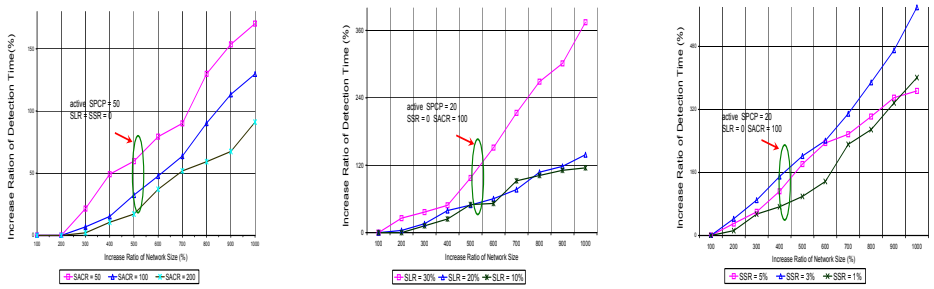

Figure 6: The Impact of Network Size (a) Without Symptom loss and spurious symptoms (b) With symptom loss (c) With Spurious symptoms

also considers the following parameters: Initial Passive Period (IPP); Symptom Active Collecting Rate ($SACR$); Symptom Passive Collecting Rate ($SPCR$); Symptom Loss Ratio (SLR); Spurious Symptom Ratio (SSR); Fidelity Threshold $FD_{THRESHOLD}$.

The major contribution of this work is to offer an efficient fault reasoning technique that provides accurate results even in worst cases like when symptom passive collecting rate ($SPCR$) is low, and/or symptom loss ratio (SLR) and spurious symptom ratio (SSR) are high. We show how these factors affect the performance (T) and accuracy (α and β) of our approach and passive fault reasoning approach.

4.1 The Impact of Symptom Loss Ratio

Symptom loss hides fault indications, which negatively affects both accuracy and performance of fault localization process. In order to study the improvement on both the performance and the accuracy of AIR approach, we fix the value of spurious symptom ratio ($SSR = 0$), the initial passive period (IPP = $10sec$), symptom active collecting rate ($SACR$ = 100 symptoms/sec) and symptom passive collecting rate ($SPCR$ = 20 symptoms/sec). In this simulation, we use SLR value that varies from 10% to 30%. With the increase of symptom loss ratio, passive fault reasoning system becomes infeasible. Therefore, in this experiment, we had to reduce the fidelity threshold to relatively lower value based on the symptom loss ratio so the passive reasoning process can converge in reasonable time. From Fig. 5(a), in contrast to passive approach, AIR system can always reach relatively high fidelity threshold with average performance improvement of 20% to 40%. Hence, when SLR increases, the advantage of active fault reasoning in the performance aspect is more evident. In addition to performance improvement, AIR approach shows high accuracy. With the same settings, Fig. 5(b) and (c) show that active approach gains 20-50$\%$ improvement of detection rate and 20-60$\%$ improvement of false detection rate, even with much different fidelity criteria over the passive reasoning approach.

4.2 The Impact of Network Size

In this section, we examine the scalability of AIR when network size and number of symptoms significantly increase. To show this, we measure AIR detection time under dif-

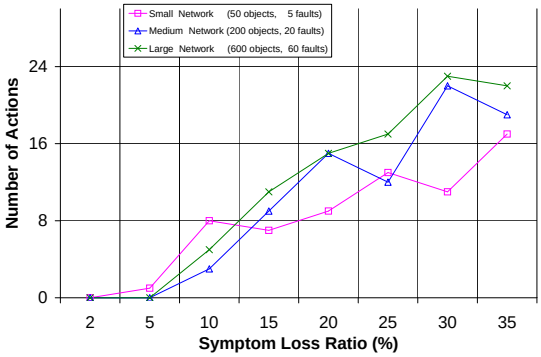

Figure 7: Intrusiveness Evaluation

ferent scenarios: (1) without symptom loss and spurious symptom (Fig. 6(a)); (2) with symptom loss only (Fig. 6(b)), and (3) with spurious symptoms only (Fig. 6(c)). In all three cases, when the network size increases 10 times (from 100% to 1000%), the detection time has slowly increased by 1.7 times (170%) and 3.7 times (370%) and 5.8 times (580%) in Fig. 6(a), (b) and (c) respectively. This shows that even in the worst case scenario (Fig. 6(c)), the growth in network size causes a slow linear increases on AIR performance.

4.3 The Impact of Symptom Loss on AIR Intrusiveness

AIR intrusiveness is measured by the number of total actions performed to localize faults. As shown in Section 3, the intrusiveness of AIR was algorithmically minimized by (1) considering the fault hypothesis of high credibility, and (2) selecting the minimum-cost actions based on the greedy algorithm described in Section 3.3. We also conducted experiments to assess the intrusiveness (i.e., action cost) when the loss ratio increases. Loss ratio and network size are the most significant factors that might affect the intrusiveness of AIR. Fig. 7 shows that, with different scale of network sizes and prior fault probability as high as 10%, the number of actions required for fault localization increases slow linearly (from 1 - 22) even when the loss ratio significantly increases (from 2%-35%). For example, in large-scale network of size 600 objects and fault rate is 60 faults per iteration, the number of action performed did not exceed 0.37 action/fault ratio. In addition, AIR was deliberately designed to give the user the control to adjust the intrusiveness of active probing via configuring the following fault reasoning parameters: fidelity threshold, IPP and action coverage.

5. RELATED WORK

Many proposed solution were presented to address fault localization problem in communication networks. A number of these techniques use different causality model to infer the observation of network disorder to the root faults. In our survey, we classify the related work into two general categories:

Passive Approach. Passive fault management techniques typically depended on monitoring agents to detect and report network abnormality using alarms or symptom events.

These events are then analyzed and correlated in order to reach the root faults. Various event correlation models were proposed including rule-based analyzing system [11], model-based system [13], case-based diagnosing system and model traversing techniques. Different techniques are also introduced to improve the performance, accuracy and resilience of fault localization. In [7], a model-based event correlation engine is designed for multi-layer fault diagnosis. In [2], coding approach is applied to deterministic model to reduce the reasoning time and improve system resilience. A novel incremental event-driven fault reasoning technique is presented in [3] and [4] to improve the robustness of fault localization system by analyzing lost, positive and spurious symptoms.

The techniques above were developed based on passively received symptoms. If the evidence (symptoms) are collected correctly, the fault reasoning results can be accurate. However, in real systems, symptom loss or spurious symptoms (observation noise) are unavoidable. Even with a good strategy [4] to deal with observation noise, those techniques have limited resilience to noise because of their underlying passive approach, which might also increase the fault detection time.

Active Probing Approach. Recently, some researchers incorporate active probings into fault localization. In [6], an active probing fault localization system is introduced, in which pre-planned active probes are associated with system status by a dependency matrix. An on-line action selection algorithm is studied in [5] to optimize action selection. In [9], a fault detection and resolution system is proposed for large distributed transaction processing system.

Active probing approach is more efficient in locating faults in timely fashion and more resilient to observation noise. However, this approach has the following limitation:

- Lack of integrating passive and active techniques in one framework that can take advantage of both approaches.
- Lack of a scalable technique that can deal with multiple simultaneous faults.
- Limitation of some approaches to track or isolate intermittent network faults and performance related faults because they solely depend on the active probing model.
- The number of required probes might be increased exponentially to the number of possible faults ([5]).

Both passive and active probing approaches have their own good features and limitations. Thus, integrating passive and active fault reasoning is the ideal approach. Our approach combines the good features of both passive and active approaches and overcome their limitations by optimizing the fault reasoning result and action selection process.

6. CONCLUSION AND FUTURE WORK

Fault localization technique plays a critical role in managing and maintaining large scale communication networks. How to improve efficiency and accuracy of fault localization technique and relieve heavy burden of system administrators continues to be a very interesting research topic. In this paper, a novel technique called ACTION INTEGRATED FAULT REASONING or AIR is presented. This technique is the first to seamlessly integrate passive and active fault reasoning in order to reduce fault detection time as well as improve the accuracy of fault diagnosis. AIR approach is designed to minimize the intru-

siveness of active probing via enhancing the fault hypothesis and optimizing the action selection process. Our simulation results show that AIR is robust and scalable even in extreme scenarios such as large network size and high spurious and symptom loss rate.

In our future work, we will study the use of positive symptoms in AIR, and optimize the fault reasoning algorithm to reduce the hypotheses searching time. In addition, we will investigate the automatic creation of the Action-Symptom correlation matrix from the network topology and high-level service specifications.

References

[1] EHAB AL-SHAER, YONGNING TANG. QoS Path Monitoring for Multicast Networks, *Journal of Network and System Management (JNSM)*, (2002).

[2] S. KLIGER, S. YEMINI, Y. YEMINI, D. OHSIE, AND S. STOLFO. A coding approach to event correlation, *Proceedings of the Fourth International Symposium on Intelligent Network Management*, (1995).

[3] M. STEINDER AND A. S. SETHI . Increasing robustness of fault localization through analysis of lost, spurious, and positive symptoms, *In Proc. of IEEE INFOCOM*, (New York, NY, 2002)

[4] M. STEINDER AND A. S. SETHI . Probabilistic Fault Diagnosis in Communication Systems Through Incremental Hypothesis Updating, *Computer Networks Vol. 45, 4 pp. 537-562*, (July 2004)

[5] I. RISH, M. BRODIE, N. ODINTSOVA, S. MA, G. GRABARNIK. Real-time Problem Determination in Distributed Systems using Active Probing, *IEEE/IFIP (NOMS)*, (Soul, Korea, 2004).

[6] BRODIE, M., RISH, I. AND MA, S.. Optimizing Probe Selection for Fault Localization, *IEEE/IFIP (DSOM)*, (2001).

[7] K. APPLETY et al. Yemanja - a layered event correlation system for multi-domain computing utilities, *Journal of Network and Systems Management*, (2002).

[8] THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD L. RIVEST AND CLIFFORD STEIN. Introduction to Algorithms, Second Edition, *The MIT Press* ,

[9] J. GUO, G. KAR AND P. KERMANI. Approaches to Building Self Healing System using Dependency Analysis, *IEEE/IFIP (NOMS)*, (Soul, Korea, 2004).

[10] J. PEARL. Probabilistic Reasoning in Intelligent Systems, *Morgan Kaufmann Publishers*, (San Francisco, CA (1988)).

[11] G. LIU, A. K. MOK, AND E. J. YANG. Composite events for network event correlation, *Integrated Network Management VI, pages 247260*, (Boston, MA, May 1999).

[12] K. HOUCK, S. CALO, AND A. FINKEL. Towards a practical alarm correlation system, *Integrated Network Management IV*, (Santa Barbara, CA, May 1995).

[13] G. JAKOBSON AND M. D. WEISSMAN. Alarm correlation, *IEEE Network, pages 5259*, (Nov. 1993).

[14] I. KATZELA AND M. SCHWARTZ. Schemes for fault identification in communication networks, *IEEE Transactions on Networking, 3(6)*, (1995).

[15] S. A. YEMINI, S. KLIGER, E. MOZES, Y. YEMINI, AND D. OHSIE. High speed and robust event correlation, *IEEE Communications Magazine, 34(5):8290*, (1996).