Active Fault Reasoning in Communication Networks

Yongning Tang and Ehab Al-Shaer
School of Computer Science,
Telecommunications and Information Systems
DePaul University, Chicago, USA
{ytang,ehab}@cs.depaul.edu

Abstract

Different fault reasoning techniques are used in fault localization for either deterministic or probabilistic fault causality model. Symptom-Fault map is commonly used to describe Symptom-Fault causality in fault reasoning. However lost and spurious symptoms severely affect both performance and accuracy of fault reasoning. In this paper, we propose an extended Symptom-Fault-Action model to incorporate actions into fault reasoning process to tackle the above problem. 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.

1 INTRODUCTION

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} [2]. Symptom-Fault causality graph provides a vector of correlation likelihood measures, $p(s_i|f_i)$, to bind a fault, f_i, to a set of its symptoms, S_{f_i}. If $p(s_i|f_i) = 0$ or 1, then the Symptom-Fault correlation has a deterministic model, otherwise (i.e. when $0 < p(s_i|f_i)$) it is a probabilistic model.

Two approaches are commonly used in fault reasoning and localization: passive diagnosis ([1], [2], [3], [7], [4]) and active probing ([6], [5] and [9]). In this paper, we propose 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, when passive reasoning is not sufficient, the optimal probing actions are selected in order to discover the most critical symptoms that could have been lost or corrupted during passive fault reasoning. Thus, our approach significantly improves the performance of fault localization while minimizing the intrusiveness of active fault reasoning.

Fault localization techniques are required to identify faults not only accurately, but also in a timely fashion. Thus, the performance of fault localization depends on the rate and accuracy of the symptom collection and analysis. Many faults might cause serious damages if they are not discovered and resolved promptly. For example, significant interruption of a web server of e-commerce applications may directly result in losing customers. Our fault localization technique, AIR, was developed to satisfy the following objectives:

- High-performance and low latency fault detection
- Accurate root cause analysis
- Handling deterministic and probabilistic causality models

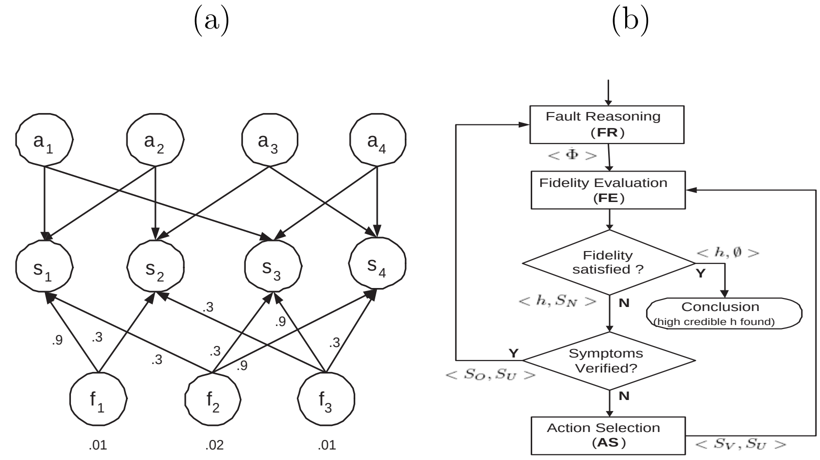

Figure 1: (a) Action-Symptom-Fault Model (b)Active Action Integrated Fault Reasoning

- Scalability for large number of managed objects and faults
- Minimal network intrusiveness
- Adjustability to satisfy the user and network requirements

The paper is organized as follows. In section 2, we discuss our research motivation and formalize the problem. In section 3 we describe the components of AIR and all related algorithms. In Section 4 we present a simulation study. In Section 5 related work is discussed. In section 6, we wrap up the paper with 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 reported using passive fault management systems if agents are configured to report abnormal system conditions or symptoms such as high average packet drop ratio. On the other hand, symptoms can be lost due to noisy unreliable communications channels, or corrupted due to spurious (untrue) symptoms, that might be 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 active actions into traditional Symptom-Fault model, we propose an extended Symptom-Fault-Action model as shown in Fig. 1(a). 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 [9] and EPP [10]. 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 like 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(a). 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 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. The basic Symptom-Fault-Action model can be described as the following:

- For every action, associates an action vertex a_i, $a_i \in A$;
- For every symptom, associates a symptom vertex s_i, $s_i \in S$;
- For every fault, associates 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 most two important factors for evaluating fault localization techniques. Performance is measured by fault detection time T, which is the time between receiving the trouble tickets (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. We will show in our simulation study in Section 4 that our approach shows a significant improvement in performance and accuracy over the passive approach.

3 ACTIVE INTEGRATED FAULT REASONING

The Active Integrated Fault Reasoning (AIR) process (Fig. 1(b)) 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, $h_i \in \Phi$, 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_V, S_U, S_O 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

Fault Reasoning is the process of searching for the best fault explanation of the observed symptoms. Symptom-Fault causality map is commonly used fault reasoning model. For each fault f_i, Symptom-Fault causality map provides a vector of correlation likelihood measures $p(s_j|f_i)$ associated with correlated symptom set S_{f_i}, where $s_j \in S_{f_i}$. S_{f_i} includes all symptoms caused by f_i, which implies the occurrence of this fault. In the fault reasoning process, it is a commonly assumed that the probability of multiple faults happening simultaneously is low.

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 following, we use S_{O_i} to denote the set of observed symptoms so far.

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 $f_i \in F_{s_i}$ 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(a), 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.

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

Algorithm 1 Fault Reasoning Algorithm $FR(S_O)$

Input: observed symptoms S_O

Output: fault hypothesis set Φ

Initialize: $F_C \leftarrow \emptyset, h \leftarrow \emptyset, \Phi \leftarrow \emptyset$

1: **for all** $f_i \in F$ **do**

2:   **if** $S_{f_i} \cap S_O \neq \emptyset$ **then**

3:     $F_C \leftarrow F_C \cup \{f_i\}$

4:   **end if**

5: **end for**

6: $\Phi = HU(h, S_O, F_C)$

7: return $< \Phi >$

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

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

$\hat{p}(f_i|s_i)$ shows the relative probability of f_i happening by observing s_i. For example, in Fig. 1(a), 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.

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

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 f_i 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 (Algorithm 4).

Algorithm 1 describes the fault reasoning algorithm. First it finds the fault candidate set F_C including all faults that can explain at least one symptom $s_i \in S_O$ (lines 1-4), then it calls the function $HU()$ (line 6) 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, in F_C, algorithm 2 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 $S_K = S_O$ (Algorithm 1 line 6). If multiple faults have same contribution, multiple hypotheses will be generated (lines 13-17). The searching process (HU) will recursively

Algorithm 2 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** $f_i \in F_P$ **do**
3:   **if** $C(f_i) > c_{max}$ **then**
4:     $c_{max} \leftarrow C(f_i)$
5:     $F_S \leftarrow \emptyset$
6:     $F_S \leftarrow F_S \cup \{f_i\}$
7:   **else**
8:     **if** $C(f_i) = c_{max}$ **then**
9:       $F_S \leftarrow F_S \cup \{f_i\}$
10:     **end if**
11:   **end if**
12: **end for**
13: **for all** $f_i \in F_S$ **do**
14:   $h_i \leftarrow h \cup \{f_i\}$
15:   $S_{K_i} \leftarrow S_K - S_{O_i}$
16:   $F_{P_i} \leftarrow F_P - \{f_i\}$
17: **end for**
18: **for all** $S_{K_i} = \emptyset$ **do**
19:   **if** $S_{K_i} = \emptyset$ **then**
20:     $\Phi \leftarrow \Phi \cup \{h_i\}$
21:   **end if**
22: **end for**
23: **if** $\Phi \neq \emptyset$ **then**
24:   return $< \Phi >$
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**

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).

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. 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.

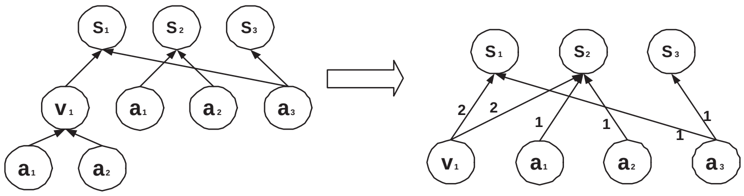

Figure 2: Symptom-Action Bipartite Graph

- For deterministic model:

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

- For probabilistic model:

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

Algorithm 3 Fidelity Evaluation $FE(\Phi)$

Input: Φ and S_O

Output: the hypothesis with highest fidelity value, corresponding unobserved symptom set S_N

1: $fd_{max} = FD(h_1)$
2: **for all** $h_i \in \Phi$ **do**
3:    $fd = FD(h_i)$
4:    **if** $fd \geq fd_{max}$ **then**
5:       $fd_{max} = fd$; $j = i$
6:    **end if**
7: **end for**
8: **if** $fd_{max} < FD_{THRESHOLD}$ **then**
9:    $S_N \leftarrow \{\bigsqcup_{f_i \in h_j} S_{f_i}\} - S_O$;
10: **else**
11:    $S_N \leftarrow \emptyset$
12: **end if**
13: return $< h_j, S_N >$

Fidelity evaluation determines the rank of each hypothesis and decides whether the fault reasoning result is satisfactory or not. Algorithm 3 evaluates each hypothesis using fidelity evaluation function and decides if the result is satisfactory by comparing to the pre-defined threshold value $FD_{THRESHOLD}$. If an acceptable hypothesis that matches the fidelity threshold exists, the FE algorithm returns this hypothesis (lines 2-7, 11). Otherwise, the best available hypothesis and a non-empty set of symptoms (S_N) to be verified are returned (line 9) in order to reach a satisfactory hypothesis in the next iteration.

3.3 Action Selection Heuristic Algorithm

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. The goal of the Action Selection algorithm is to select the actions that cover all symptoms, S_N, in the graph 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*.

The weighted set-covering is an NP-complete problem. Thus, we developed a heuristic greedy set-covering approximation algorithm (Algorithm 4) to solve this problem. The main idea of the Algorithm 4 is simply selecting first the action (a_i or v_i) that has the maximum ratio of the *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 (Lines 3-5). 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 (Line 6). This search continues to find the next action, $a_i \in A_c$, that has the maximum ratio R_i until all symptoms are covered (i.e., S_N is empty) (Line 2). Thus, intuitively, this algorithm appreciates actions that have more symptoms 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.

In order to control the trade-off between the searching time and accuracy (i.e., finding close to optimal solution), we use this greedy algorithm until size of S_N becomes smaller than a threshold (G) after which we use an exhaustive search technique to improve accuracy (line 7). The function performAction(A_S) executes the selected actions from A_f (Line 13) and reports the occurred (existing) symptom set S_V and the not-occurred (non-existing) symptom set S_U (Lines 13-14).

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

Algorithm 4 Action Selection $AS(S_N)$

Input: a set of unobserved symptoms S_N

Output: final selected action set A_f, verified occurred symptom set S_V and unoccurred symptom set S_U

Initialize: $A_S \leftarrow \emptyset$, $S_V \leftarrow \emptyset$, $S_U \leftarrow \emptyset$

find A_C containing actions that can verify at least one symptom in S_{N_i}
while $S_N \neq \emptyset$ **do**
    find $a_i \in A_C$ with maximum covering ratio R_i
    $A_C = A_C - \{a_i\}$
    $A_s \leftarrow a_i$
    $S_N \leftarrow S_N - S_{a_i}$
    if $|S_N| < G$ **then**
        do exact searching for optimized minimum-size action set
    else
        continue
    end if
end while
$< S_V, S_U >= performAction(A_S)$
return $< S_O >$

reach to root faults.

Algorithm 5 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. IPP is used to control passive symptom collecting period before initiating actions to avoid unnecessary actions in case the symptom passive collecting rate ($SPCR$) is relatively low.

4 SIMULATION STUDY

In this section, we describe our simulation study to evaluate the proposed Action Integrated fault Reasoning (AIR) technique. We conducted a series of experiments to measure how our approach 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

Algorithm 5 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

accuracy parameters.

In our simulation study, the number of monitored network objects, D, such as web servers and routers, 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 are uniformly distributed $p(s_i|f_j)$ in ranges [0.001, 0.01] and (0, 1] respectively. Our simulation model 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 $FD_{THRESHOLD}$.

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 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%. From Fig. 3(a), on contrast to passive approach, AIR system can always reach relatively high fidelity thresh-

(a) (b) (c)

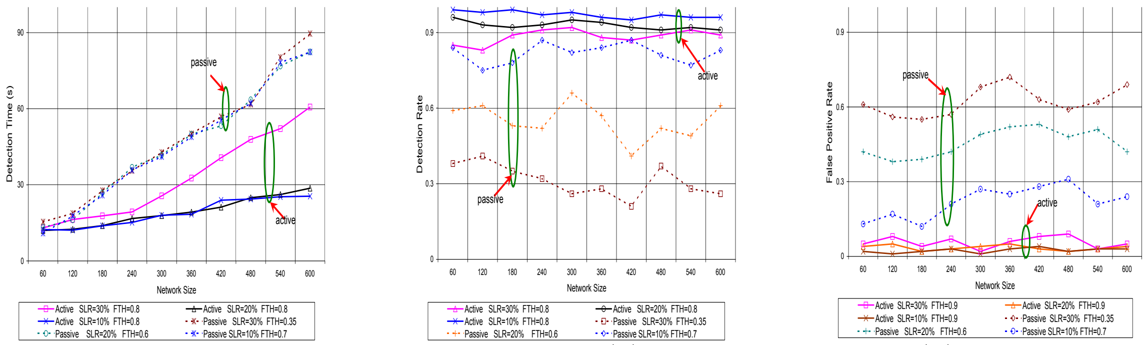

Figure 3: The Impact of Symptom Loss Ratio (a) Detection Time T (b) Detection rate α (c) False positive rate β

(a) (b) (c)

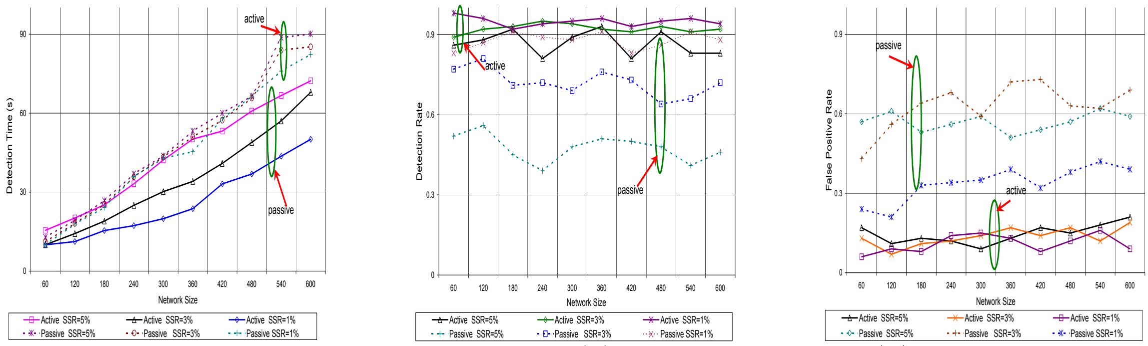

Figure 4: The Impact of Spurious Symptoms (a) Detection time T (b) Detection rate α (c) False positive rate β

old ($FD_{THRESHOLD} = 0.8$) with average performance improvement of 20% to 40%. Hence, when SLR getting bigger, 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. 3(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 Spurious Symptoms

The spurious symptoms are also regarded as observation noise, which could seriously affect fault reasoning because they provide misleading information rather than losing information. To isolate the impact of spurious symptoms, we set $SLR = 0$ and fix IPP with $10sec$ and $SACR$ with 100 symptoms/sec. The relative signal-noise ratio can be calculated as $SNR = \frac{1-SSR}{SSR}$ if $SLR = 0$. Fig. 4(a) shows that on average AIR will have 10-20% improvement of the performance over the passive approach even with high fidelity value. With the same experiment settings, in Fig. 4(b) and (c), AIR shows accuracy improvement of 10-50% for the detection rate and 10-40% for the false positive rate over the passive approach.

5 RELATED WORK

Many proposed solution were presented to address fault localization problem in communication networks. 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. 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 [1], 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.In real systems, symptom loss or spurious symptoms (observation noise) are unavoidable. Even with good strategy ([2] and [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.

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: 1)Lack of integrating passive and active techniques in one framework that can take advantage of both approaches; 2)Lack of a scalable technique that can deal with multiple simultaneous faults; 3) Limitation of some of these approaches to track or isolate intermittent network faults and performance related faults because they solely depends on the active probing model; 4) 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

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 intrusiveness 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 would investigate the automatic creation of the Action-Symptom correlation matrix from the network topology and high-level service specifications.

References

[1] 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).

[2] M. Steinder and A. S. Sethi . Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system, *IEEE International Conference on Computer Communications and Networks (ICCCN)*, (Scottsdale, AR, 2001), pp. 374-379.

[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] S. Thomas H. Cormen et al. Introduction to Algorithms, *The MIT Press*, (2001), Second Edition.

[9] Ehab Al-Shaer, Yongning Tang. QoS Path Monitoring for Multicast Networks, *Journal of Network and System Management (JNSM)*, (2002).

[10] A. Frenkiel and H. Lee.. A Framework for Measuring the End-to-End Performance of Distributed Applications, *In Proceedings of Performance Engineering Best Practices Conference, IBM Academy of Technology*, (1999).

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

[12] K. Houck, S. Calo, and A. Finkel. Towards a practical alarm correlation system, *Integrated Network Management IV*, (Santa Barbara, CA, May 1995).