Centralized vs Distributed Fault Identification in Communication Networks

I. Katzela *
CTR-Columbia University ‡
Rm 801 Shapiro Bld.
530 west 120th street
NewYork, NY 10027-6699
irene@ctr.columbia.edu
tel:(212)854-7378

A.T. Bouloutas †
First Bank of Boston
Treasure Mail Stop 01-10-09
100 Federal Street
MO BOX 2016
Boston MA 0216
tel:(617)434-0534

S.B. Calo
IBM Corporation
T.J. Watson Research Center,
P.O. Box 704, Yorktown Heights, NY 10598
calo@watson.ibm.com
tel:(914)784-7514

June 29, 1994

Abstract

In this paper we compare the performance of fault identification and alarm correlation schemes for communication networks. Our model assumes a number of management centers, each responsible for a logically autonomous part of the whole telecommunication network. We briefly present three different fault localization approaches, namely "Centralized", "Decentralized" and "Distributed" fault localization and we compare their performance with respect to the computational effort each requires and the accuracy of the solution that each provides.

*Work done during the author's intership at IBM Corp. T. J. Watson Research Center, NY, Summer 93.
†Work done during the author was with the IBM Corp.T. J. Watson Research Center, NY.
‡CTR - Center of Telecommunications Research

1 Introduction

As networks evolve and become larger and more complex, the need for advanced network management capabilities become more critical. Human operators will not be able to keep up with the day to day requirements for management without automation. One of the most important areas that needs automation is fault management, and closely allied to it is the problem of alarm correlation and fault identification. Usually, a single fault in a large network results in a number of alarms, and it is not always easy to identify the primary source(s) of failure. The problem becomes even worse when several faults occur coincidentally in the telecommunication network.

In the past a number of researchers addressed the problem of fault localization in large telecommunication networks [1], [11], [10], [9], [4], [7], [8], [5]. These efforts resulted in a number of proposed methods for fault identification and alarm correlation. Most of these methods focus on centralized algorithms for fault identification. However, the growth in size and complexity of communication networks may require the partitioning of the management environment into a number of management domains in order to meet organizational and performance requirements. The partitioning may be directed by geographical, technological or organizational requirements, and might be static or dynamic. This transition from a centralized management environment to a distributed one will require the development of distributed algorithms for alarm correlation and fault localization.

The distributed model of the system assumes the existence of static, disjoint, logically autonomous domains each managed by a single management process. Each manager communicates with other managers in a peer to peer fashion, and there may or may not be a central manager that oversees all the domain managers. A distributed fault management system will be able to shield parts of the network management system from information that is not locally useful, a very important function as management centers tend to overflow with information. However, problems that involve objects in more than one domain will have to be resolved collectively by many manager processes in a distributed fashion. This introduces a number of problems that make the design of distributed fault management solutions a challenging task.

The fault management process can be divided into three stages: alarm correlation, fault identification, and testing. The first two stages, usually referred to as the fault localization process, correlate the fault indications (alarms) received from the managed objects and propose various fault hypotheses. In the third stage each of the proposed hypotheses is tested in order to localize the fault precisely. The fault localization process is important because the speed and accuracy of the fault management process are heavily dependent on it.

The objective of any fault management system is to minimize the time to localize a fault. The time to localize a fault is the sum of the time to propose possible fault hypotheses (fault identification) and the time to perform testing in order to verify these hypotheses. The time to propose various hypotheses of faults is affected by the fault identification algorithm. The time required for testing is affected by the number of managed objects that must be tested. Thus, if the network management system is able to identify the source of a fault, the minimum number of tests will have to be performed. This paper is organized as follows: in section 2 we define the problem of distributed fault localization and present a suitable model for the system; in section 3 we present three different approaches for distributed fault localization; in section 4 we compare the proposed approaches with respect to the computational effort each requires and the accuracy of the solution each provides; and finally, section 5 concludes with a summary of the results.

2 Model of the System

We assume a distributed approach to managing communication networks [6] [2]. Each communication network is partitioned into a number of static, disjoint, logically autonomous management domains. Each domain is managed by a single management process [1] which is responsible for it. The managed objects in a domain may or may not be visible to managers of other domains. Hence, each manager has a limited view of the status of other management domains; and, each domain manager has partial and incomplete information about the state of the network. However, managers from different domains can communicate and exchange information about the status of their domains using a peer to peer type of communication. Each domain manager has adequate knowledge about adjacent managers and domains so that communication can be established and messages passed.
Faults manifest themselves as alarms and alerts which are emitted by the managed objects affected by the faults. Alarms are communicated to the managed object's management process, which is responsible for identifing the primary source(s) of failure and eventually correcting the fault(s). Each alarm emitted by a managed object represents the fault from the point of view of the object that emitted the alarm, and therefore corresponds to partial information about the fault. It is the responsibility of the fault identification system to collect all these partial views of a fault, correlate the information, and infer the real cause(s) of the fault. It is not unusual for an alarm to appear in a managed object belonging to a particular management domain and to indicate a fault in another managed object in a different domain. Since alarms cross management domains, management centers have to collaborate in order to infer the real state of the system.
Note that throughout the paper we assume that faults affect only managed objects but do not affect managers, information transfer processes of management systems, or other parts of the Telecommunication Management Network (TMN). This is a reasonable assumption and it stems from the fact that usually TMN systems have much stricter reliability requirements than the rest of the network.

[1] In this paper we will use the terms management process, manager and management center interchangeably.

Before we proceed it is essential to examine the structure of the alarms. Each alarm is characterized by the *domain of the alarm* which is defined as the set of all independent managed objects that could have caused the alarm - in other words, all the managed objects that might be at fault. Note that the domain of an alarm should not be confused with the domain of a management center. The domain of an alarm depends both on the semantics of the alarm and the topology of the communication network. It is the management centers' responsibility to find the domain of a received alarm before proceeding to the fault localization process.

3 Fault Localization Approaches

Assume that at a given moment in time a number of alarms appear in a communication network. The objective is to design algorithms that are able to find the *"best"* explanation of the received alarms, i.e., the managed object or the set of managed objects that could have been at fault and caused the alarms. In principle all the managed objects that appear in the domain of a received alarm constitute a possible explanation of it. If two or more alarms share an intersection, these alarms should be examined together because it is more probable that they are caused by the same set of faults. That is the reason why we introduce the notion of a cluster of alarms. A *cluster* of alarms is defined as a set of alarms that have intersecting domains. An alarm belongs to a cluster if it has an intersection with some alarm that belongs to this cluster. Two different clusters have no intersection. Note that a cluster may span more than one management domain as the alarms that comprise the cluster span more than one domain [6], [2].

Each cluster of alarms may have a number of explanations. The fault identification algorithm should be able to choose the *"best"* one among the possible ones. The best one is the most probable one. One way to find the most probable explanation would be to associate a probability of failure with each managed object. In this case the most probable explanation would be the set of managed objects whose combined probability of failure is maximum. Instead of assigning a probability of failure to each managed object we can associate an *"information cost"* which is defined as the negative of the logarithm of the probability of failure for the managed object. For independent faults, the information costs are equivalent to probabilities, but working with them has certain advantages. One of these is the additivity property: if we assume that the probability of failure of each of the managed objects is independent of the state of the other managed objects, then the combined probability of a set of managed objects would be the product of probabilities of each of the elements in the set, while the corresponding information cost will be the sum of the information costs of the elements in the set. If we choose to work with information costs then the *"best"* solution (most probable) will be the set of managed objects whose sum of information costs is minimum [6], [2].

Before we proceed to examine distributed fault localization approaches, we assume that there exists a centralized algorithm that can find the most likely errors in a set of managed objects, given a set of alarms and the information costs associated with each managed object. As was shown in [3] the fault identification problem is NP-Complete. Thus, in general there is no polynomial algorithm that gives the exact solution. One could then either construct a polynomial algorithm that gives an approximate solution, or a polynomial algorithm that gives the correct solution with some probability. In Appendix A, we present a possible probabilistic algorithm which finds an exact solution if the number of faults in the system is less than k, a parameter. We represent this algorithm by $G_p(A, N, k)$ where A is the received alarm cluster, N is the number of managed objects associated with A, and k is the maximum number of concurrent failures that can be identified by the algorithm. The probabilistic algorithm fails to give a solution with probability $1-q$ which is equal to the probability that there are more than k faults in the system. Hence:

$$1 - q = P_r(\text{algorithm fails}) \quad = \quad P_r(> k \text{ faults in the network}) \leq 1 - \sum_{i=0}^{k} b(i; N, p) \tag{1}$$

Each managed object has a probability of failure assigned to it, p is the maximum of all such probabilities for the managed objects associated with the received alarm cluster A, and $b(i; N, p)$ is the probability of N Bernoulli trials, with probability p of success and i successes. The details of the algorithm are given in Appendix A.

Since each management center can resolve problems that affect managed objects in its domain we will only examine the case where faults affect objects in many management domains. For simplicity we assume first that the entire network consists of two domains. A generalization of the results for multiple domains will be discussed in appendix C. We also, without loss of generality, assume that there is only one cluster of alarms that crosses the boundary between domains.

3.1 Centralized Localization

The first approach, namely *Centralized Localization*, assumes the existence of a central manager that oversees all the domain managers and has a global view of the network. Problems that affect more than one domain could be resolved directly by the central manager as if there were no domain managers. In other words, if the received alarm cluster spans more than one domain, then the domain managers take no action and the central manager uses a centralized algorithm like $G_p(A, N, k)$ to identify the failure. The "Centralized Localization" approach always guarantees to output the optimum explanation of the received alarms. It fails to output an explanation of the received alarms with probability $1-q$ given by (1).

3.2 Decentralized Localization

The second approach, namely *Decentralized Localization*, assumes the existence of a central manager that oversees all the domain managers. Problems that affect more than one domain could be resolved in a collaborative way between the central manager and the domain managers. Unlike the first approach, the second one does not require extensive involvement of the central manager.

Let us assume that L alarms are received and all belong to the same cluster A which is associated with N managed objects that might fail. Assume m of the L alarms cross the boundary between the domains. These m alarms might have been produced by a fault in either domain and there is no a-priori information as to whether an alarm that crosses the boundary is explained by a fault in the first domain or by a fault in the second domain, or both. If this information were available, each of the domain managers could have applied the centralized probabilistic algorithm to objects and alarms in its domain and solved the problem locally. Then the combination of the two partial solutions would give the optimum global solution. Generally, this information is not available a priori. Here we present a decentralized fault identification that quarantees to output the optimum global solution.

Each domain manager calculates, using perhaps the $G_p(A, N, k)$ algorithm, 2^m optimum solutions, one for each of the possible explanations of the m alarms that cross the boundary. If m alarms cross the boundary between the two management domains there are 2^m possible explanations for these m alarms depending on whether faults in domain one or domain two explain the alarms. Two partial solutions (one from each manager) can be combined to produce a global solution only if these two partial solutions are compatible. Two solutions are compatible if all the alarms received by both domains are explained in the final solution. The central manager receives the 2^m optimum solutions from the two domain managers and finds the compatible ones. Then it selects the compatible global solution of minimum information cost. As one can easily verify the above described procedure is able to identify the optimum solution given that the two management domains can find the optimum solution for managed objects and alarms in their respective domains [6]. Finally, since each domain manager uses the probabilistic algorithm for fault identification there is a probability $1 - q'$ (which will be calculated in section 4.1) that the *Decentralized Localization* fails to output a solution.

3.3 Distributed Localization

The third approach, namely *Distributed Localization* , does not assume the existence of a central manager. The area of the network is divided into two domains, each managed solely by a single domain manager. This strategy tries to find the faults from the point of view of each of the two domain managers without the use of a central manager. Let us examine the problem from the point of view of domain manager one. For each alarm that cross the boundary, domain manager one would like to associate a probability that this alarm is explained by *domain two*. One way to do that would be to represent all the managed objects that belong to domain two with a *proxy node*. The *proxy node* represents all the managed objects in domain two that could have caused the alarm. Failure of the proxy node would indicate that some managed objects in domain two have failed. If one could associate a probability of failure with the proxy node then the management process of domain one could use the probabilistic algorithm $G_p(A, N, k)$ to solve a centralized problem in a new expanded domain that includes the managed objects in its domain plus m proxy nodes, one for each alarm that crosses the boundary between the two domains. Here all the alarms that cross the boundary are treated as regular alarms. Once the algorithm has output the optimum solution there will be some alarms that are explained by regular nodes in domain one and alarms that are explained by proxy nodes. The alarms that are explained by proxy nodes are the alarms that are not explained by domain one and are hopefully explained by domain two. The global solution is the one that includes all the regular nodes that appear in the optimum solutions of the two domain managers. Note that by using this algorithm we impose a penalty on domain one for every alarm that is not explained and is left to be explained by a fault in domain two.

The exact probability of failure for a proxy node is difficult to find since it depends on all the managed objects and all the alarms in the cluster. In general the exact calculation of the probability of a proxy node is an NP-complete problem [6]. The best we can achieve is an estimation of the probability of failure for a *proxy node*. The estimated probability for a *proxy node* differs from the exact one by an estimation error. As a result of the estimation errors the distributed localization approach does not always guarantee an optimum global solution [2].

4 Performance Comparison

All the approaches presented in the previous sections are heuristic methods to solve a problem which is inherently NP-Complete. In the process of designing a heuristic algorithm for an NP-hard problem the engineer must answer a number of questions such as: Can some heuristic beat the exponential explosion? How much speed can be gained at the cost of lowering the solution quality? Any resulting heuristic choice is characterized by a set of parameters or attributes and based on these attributes the choice would be suitable or unsuitable for a class of encountered problems. Most importantly we would like to understand what attributes make a candidate heuristic choice suitable or unsuitable for a given class of problems. Conversely, if the attributes are given, which is the class of problems that the candidate choice is capable of solving.

Towards this goal we started, in the previous sections, with the definition of a simplified abstract model for the problem environment which contains a small number of parameters required to represent the main features of the problem domain; and, now we select the performance measures and analyze the performance of the various candidate approaches as a function of the model's parameters. The goals of the analysis are to find those model parameters, if any, that have a dramatic influence on the relative performance of the candidate identification algorithms, and to identify the class of problems that each of the algorithms is suitable to solve under certain conditions. Finally, if one of the candidate methods proved to exhibit superior performance over a wide range of parameter values then it would become the first choice to be tried in actual problem settings.

As we presented in the introduction, the objective of any fault management process is to to minimize the time to localize a fault. The time to localize a fault is the sum of the time to propose possible hypotheses of faults and the time to do testing in order to verify these hypotheses. Thus, we should minimize the time to perform fault identification and the time to perform testing. The time to identify the fault is affected by the identification algorithm. Hence, the first objective is to minimize the time complexity of the identification algorithm. The second objective is to minimize the time of testing. The time of testing is affected by the number of managed objects that need to be tested. The number of tests to be performed is equal to the number of proposed hypotheses by the identification algorithm. If the management process is able to identify correctly the source of failure, the minimum number of tests is required. Thus, minimizing the number of tests is equivalent to minimizing the number of fault hypotheses, or equivalently, maximizing the accuracy of the fault identification algorithm.

Finally, the complexity of the identification algorithm for each of the previously proposed approaches is a function of the number of received alarm clusters per unit time, the number of nodes per cluster of received alarms, the number of alarms that cross the domains, and the parameter of the probabilistic algorithm that is the base of both the proposed identification schemes. On the other hand, the accuracy of each approach depends on the error in the estimations of the probabilities of failure for the managed objects associated with the received cluster of alarms.

4.1 Comparison between Centralized and Decentralized Fault Identification

Assume that we have received one cluster of alarms, A, which is associated with N managed objects that may fail. Each of these objects has probability of failure p. We would like to compare the performance of the centralized versus the decentralized approach for this network setting. For the centralized approach, the central manager process should use the probabilistic algorithm $G_p(A, N, k)$. For the decentralized approach we assume that we partition the managed system in two domains namely *Domain one* and *Domain two*. Also we assume that there are m alarms that crosses the boundaries between the two domains, and the N managed objects associated with the received cluster of alarms A are partitioned into N_1 objects in domain one and N_2 in domain two, such that $N = N_1 + N_2$. According to the decentralized approach, each domain manager uses the probabilistic algorithm

2^m times for its area in order to find 2^m optimum solutions, one for every possible interpretation of the alarms that cross the boundary. In each case, Domain one will use $G_p(A_1, N_1, k_1)$ and Domain two will use $G_p(A_2, N_2, k_2)$ to identify the optimum solution. A_1 is the set of alarms that domain manager one takes into account in this instance, A_2 is the set of alarms that domain manager two takes into account in this instance, k_1 is the number of faults that manager one can localize, and k_2 is the number of faults that manager two can localize.

The selected performance measures for comparing these approaches are *accuracy* and *time complexity*. The *accuracy* performance measure has two aspects. The first one is the difference between the information cost of the proposed solution and the optimum one. The second aspect of *accuracy* is the probability that the approach fails to give a solution. By design, both the centralized and the decentralized approaches give the optimum cost solution whenever they give a solution. Thus, in the remainder of the section we need to discuss only the second aspect of *accuracy*. Regarding accuracy, the centralized approach fails to find a solution with probability $1-q$, which is given by (1). The decentralized approach fails to find a solution with probability $1-q^{'}$ which is:

$$\begin{aligned} 1-q^{'} &= P_r(\text{Decentralized approach fails}) \stackrel{(1)}{=} \\ & \quad Pr(>k_1 \text{ faults in domain one}) + Pr(>k_2 \text{faults in domain two}) \\ &- Pr(>k_1 \text{ faults in domain one}) * Pr(>k_2 \text{faults in domain two}) = \\ & \quad \sum_{i=k_1+1}^{N_1} b(i;N_1,p) + \sum_{i=k_2+1}^{N_2} b(i;N_2,p) - \sum_{i=k_1+1}^{N_1} b(i;N_1,p) \sum_{i=k_2+1}^{N_2} b(i;N_2,p) \end{aligned} \tag{2}$$

Regarding *time complexity*, the centralized approach has complexity which is bounded by $C_{cen} = O(N^k)$ and the decentralized approach $C_{dec} = O(2^m \max(N_1^{k_1}, N_2^{k_2}) + 2^{2m})$.

It is obvious that the accuracy of the decentralized approach increases with an increase in the values of k_1 and k_2. The price to be paid is an increase in the time complexity of the approach. In terms of comparison between the approaches the problem has two parameters, accuracy and time complexity, which are dependent on each other. In order to reach some meaningful results it is appropriate to fix one of the two parameters. For a given k (number of faults that the centralized approach can localize) the accuracy of the centralized approach is fixed. We thus fix the accuracy of the two approaches, and then compare the respective complexities.

In other words, we will first define k_1 and k_2 such that the two probabilities are equal - the probability that the decentralized approach fails and the probability of failure for the centralized approach. In addition the decentralized approach should be able to identify at least as many faults as the centralized one. Hence:

$$
\begin{aligned}
P_r(\text{Decentralized approach fails}) &\leq P_r(\text{Centralized approach fails}) \\
\text{and } \; k_1 + k_2 &\geq k
\end{aligned}
\tag{3}
$$

Where the probability that the decentralized approach fails is given by (2) and the probability that the centralized approach fails is given by (1). The unknowns in (3) are the parameters k_1 and k_2. Typically it is difficult to solve such a set of inequalities. In order to simplify our analysis we will propose an approximation for calculating the parameters k_1 and k_2. Once k_1 and k_2 have been found we can easily compare the complexity of the two approaches.

4.1.1 Approximate Calculation for k_1 and k_2

As an approximation we assume that the number of faults each domain manager should localize is proportional to the number of managed objects in the alarm cluster that belong to its domain. This assumption is valid and stems from the fact that the managed objects in the system fail independently. Hence :

$$
\frac{k_1}{k_2} = \frac{N_1}{N_2} \overset{(3)}{\Rightarrow} k_1 + k_1\frac{N_2}{N_1} \geq k \Rightarrow k_1 \geq \lceil k\frac{N_1}{N} \rceil
\tag{4}
$$

The initial constraint that we have adopted is that the probability of failure for the decentralized approach should be less than or equal to the probability of failure for the centralized approach. We approximate this constraint with the following two requirements:

$$
\begin{aligned}
P_r(> k_1 \text{ faults in domain one}) &\leq \frac{N_1}{N} P_r(\text{Centralized approach fails}) \\
P_r(> k_2 \text{ faults in domain two}) &\leq \frac{N_2}{N} P_r(\text{Centralized approach fails})
\end{aligned}
\tag{5}
$$

With the proposed approximation we decompose the original complex problem in (3) into two simpler problems, one for each domain. Now it is sufficient to solve the problem only for one of the domains. The results are equivalent for the other domain. Without loss of generality we will solve the problem for domain one.

The new problem for domain one can be stated as follows: *Given a probability of failure for the decentralized approach, what is the value of the parameter k_1 that domain manager one should use in the application of the probabilistic algorithm, so that the following two inequalities hold?*

$$\begin{aligned} P_r(> k_1 \text{ faults in domain one}) &\leq \frac{N_1}{N} P_r(\text{Centralized approach fails}) \\ k_1 &\geq \lceil k \frac{N_1}{N} \rceil \end{aligned} \tag{6}$$

Which is equivalent to:

$$\begin{aligned} \overset{(1)}{\Rightarrow} \sum_{i=k_1+1}^{N_1} b(i; N_1, p) &\leq \frac{N_1}{N} \sum_{i=k+1}^{N} b(i; N, p) \\ k_1 &\geq \lceil k \frac{N_1}{N} \rceil \end{aligned} \tag{7}$$

The system of inequalities in (7) is still difficult to solve. We would like to simplify it and find a closed form solution for k_1. The left hand side of (7) is a finite sum of Bernoulli factors where the lower bound k_1 is the unknown of our problem. As we know, the Bernoulli series $b(i; N; p)$ increases with i for $i = 0, \ldots, \lceil (N+1)p \rceil$. It reaches a maximum for $i = \lceil (N+1)p \rceil$ and for $i = \lceil (N+1)p \rceil + 1$, $\ldots$, N it decreases. In order to simplify (7) we should find a simpler expression for $\sum_{i=k_1+1}^{N_1} b(i; N_1, p)$. The form of the expression depends on whether k_1 is $\lesseqgtr$ $\lceil (N_1+1)p \rceil$. We examine each case separately in Appendix B. The following table summarizes the results.

$\lceil (N_1+1)p \rceil < \lceil k \frac{N_1}{N} \rceil$	$k_1 > \max \left[\lceil \log_\beta \left(\frac{\frac{N_1}{N} \sum_{i=k+1}^{N} b(i;N,p)}{\Theta} \right) \rceil, \lceil k \frac{N_1}{N} \rceil \right]$
$\lceil (N_1+1)p \rceil > \lceil k \frac{N_1}{N} \rceil$, N_1 large	$k_1 > \max \left[\lceil (\eta - \frac{\frac{N_1}{N} \sum_{i=k+1}^{N} b(i;N,p)}{b(\lceil (N_1+1)p \rceil; N_1, p)} \rceil, \lceil k \frac{N_1}{N} \rceil \right]$
$\lceil (N_1+1)p \rceil > \lceil k \frac{N_1}{N} \rceil$, N_1 small	$k_1 > \max \left[\lceil 2\eta - \frac{\frac{N_1}{N} \sum_{i=k+1}^{N} b(i;N,p)}{b(\lceil (N_1+1)p \rceil; N_1, p)} \rceil, \lceil k \frac{N_1}{N} \rceil \right]$

Figure 1: Formulas for estimating the number of faults that domain manager one must calculate.

Note that all the formulas in Figure 1 provide an overestimation of the value of k_1. Similarly we can calculate k_2. We simply substitute, in the appropriate formulas in Table 1, N_2 for N_1 and k_2 for k_1. As an example, consider a network scenario where the received cluster of alarms is associated with $N = 100$ managed objects. Each of these objects has a probability of failure $p = 0.01$. We also assume that the managed objects in the cluster are partitioned into two domains so that $\frac{N_1}{N_2} = \frac{3}{2}$. As is shown in Figure 3, by using the previously derived formulas we get k_1 and k_2 close to their exact values. Thus, the overestimation of k_1 and k_2 is very small and the approximation works satisfactorily. A similar behavior, small overestimation, is observed for different values of p. Finally as we stated earlier, the decentralized approach should be able to localize, in total, an equal or larger number of faults. As we can observe from the curves in Figure 3, sometimes the decentralized approach has to localize as much as twice the number of faults as the centralized approach in order to achieve the same probability of failure. Such an increase in the parameters k_1 and k_2 results in increased complexity for the decentralized approach.

Once we have calculated the parameters k_1 and k_2 we are in a position to compare the time complexity between the two approaches. For a specific problem instance, where N_1 and N_2 are known, it is easy to show that the complexity of the decentralized approach is bounded by $C_{dec} = O(2^m \max((N_1)^{k_1}, (N_2)^{k_2}) + 2^{2m})$, where k_1 and k_2 are calculated with the use of the approximation above. This specific problem instance, N_1 managed objects in domain one and N_2 managed objects in domain two, could occur with probability $P_r(N; N_1, N_2)$. It is easy to show that the average complexity per domain manager per problem instance is:

$$\frac{C_{dec}^{aver}}{problem} = \sum_{i=1}^{N} P_r(N; i; N-i) i^{k_i} \tag{8}$$

Where k_i is the number of faults that the domain manager can localize in this specific instance. For fixed probability of failure for the decentralized approach, k_i can be calculated by use of Table 1. In principle, the probability of a specific problem scenario, $P_r(N; N_1, N_2)$, could follow any distribution. The value of $P_r(N; N_1, N_2)$ for a specific problem instance indicates how *likely* it is that this problem will be encountered. The concept of *likely* is related to the probability distribution that adequately represents the set of problems to be encountered. Such a probability distribution is difficult to define and analyze. Let us assume for simplicity that the partitioning of N managed objects associated with the received alarm cluster between the two domain is done randomly. Then the probability distribution of the problems to be encountered is a generalized Bernoulli distribution. Thus $P_r(N; N_1, N_2) = \frac{N!}{N_1! N_2!} = \frac{N!}{N_1!(N-N_1)!}$ and (8) becomes:

$$\frac{C_{dec}^{aver}}{problem} = \sum_{i=1}^{N} \frac{N!}{i!(N-i)!} i^{k_i} \tag{9}$$

The average complexity of the decentralized algorithm will thus be:

$$
\begin{aligned}
C_{dec}^{aver} &= 2^m \sum_{i=1}^{N} \frac{N!}{i!(N-i)!} \max\left(i^{k_{i1}}, (N-i)^{k_{i2}}\right) + 2^{2m} = \\
&= 2^{2m} \sum_{i=0}^{\lceil \frac{N}{2} \rceil} \frac{N!}{i!(N-i)!} i^{k_i} + 2^{2m} \qquad (10)
\end{aligned}
$$

We are interested in investigating the conditions under which $C_{dec}^{aver} < C_{cen}$. The average complexity of the decentralized approach depends on the values of N, k, p and m. The parameter m is the one which has the greater effect on the complexity of the decentralized approach. As we can easily observe from Figures 4 and 5, for fixed k and p the time complexity of the decentralized approach remains less than that of the centralized approach up to a certain value of m. For example in Figure 4(a) , for $k = 3$ the complexity of the decentralized approach remains strictly less than that of the centralized approach when $m \leq 4$. Similarly in 4(b) for $k = 5$ the same behavior is observed for $m \leq 7$, also in Figure 5 for $k = 7$, $m \leq 12$. Also we observe from the Figures 4 and 5 that the value of m up to which the decentralized approach exhibits less complexity that the centralized one increases with k. Finally for the same value of k the allowed number of alarms m that can cross domains so that the complexity of the decentralized approach is less than that of the centralized one increases with the probability p. For example for $k = 5$ and $p = 0.1$ in Figure 4(b) the allowed maximum m is $m = 7$, for $p = 0.01$ the maximum m is $m = 5$ in Figure 6(a), for $p = 0.001$ in Figure 6(b) $m = 2$.

4.2 Comparison between Centralized and Distributed Fault Identification

It is easy to show that the complexity of the distributed approach is always considerably less than the complexity of the centralized approach ($max(N_1^{k_1}, N_2^{k_2}) << N^k$). Hence, it remains to compare the approaches with respect to accuracy. The first aspect of accuracy is the probability of failure of the localization schemes to output a solution. Again we can select $k_1 and k_2$ for the two domains of the distributed approach so that the probabilities of failure for the centralized and distributed approaches are the same. The corresponding values of k_1, k_2 can be calculated by the use of Table 1. The calculated values of k_1, k_2 are $\leq k$. As a consequence, $max(N_1^{k_1}, N_2^{k_2}) << N^k$ and the computational complexity of the distributed sheme is always less than the complexity of the centralized one. The only remaining aspect of comparison is the difference between the solution that the distributed algorithm gives and the optimum one.

In our terminology a solution is a set of managed objects that can explain all the received alarms. A solution is characterized by its information cost which is the sum of the information costs of the managed objects that are included in the solution. Most of the time there are more than one possible solutions. All of the localization approaches discussed in this paper select among all the possible solutions the one which has the minimum information cost. The deviation of any solution from the optimum one is characterized by the difference in the information cost of the solution from the information cost of the optimum (minimum cost) solution. Unlike the centralized and the decentralized localization approaches, the distributed localization does not always guarantee that it can find the optimum solution. This deviation from the optimum solution stems from errors in the approximation of the probability of failure for the proxy nodes.

The exact probability of failure for a proxy node (and thus the exact information cost for the node) is difficult to find since it depends on all the managed objects and all the alarms in the cluster. The best we can achieve is an estimation of the information cost of a proxy node. Thus, the information cost of the proxy node differs from its exact value by an estimation error. The introduction of estimation errors might cause a difference between the solution proposed by the distributed identification algorithm and the optimum one which is given by the decentralized and the centralized approaches. It is of interest to find a bound for the difference between the information cost of the distributed solution and the optimum one. Also it is of interest to investigate how sensitive the distributed solution is to changes in the information cost of the proxy nodes.

For a given network let I^* be the information cost of the optimum solution; I_{cen}, I_{dec} and I_{dis} the information costs of the solutions that the centralized, decentralized, and distributed localization shemes give respectively. As we know, $I_{cen} = I_{dec} = I^*$. The objective is to find how close the solution that the distributed approach gives is to the optimum one. In other words we would like to find a bound for $\mid I^* - I_{dis} \mid$.

Before we answer the question: what is a good bound for $\mid I^* - I_{dis} \mid$, it is instructive to find, in general, how errors in the information cost of the managed objects affect the minimum cost solution.

4.2.1 Effect of Estimation Errors on the accuracy of the solution

Let us assume that a cluster of alarms is received. The cluster is associated with n managed objects d_i, $i = 1, \ldots, n$. For each managed object d_i an information cost w_i is assigned. Let S_i represent a set of objects such that every object in S_i belongs to the same alarm. The number s of such sets S_i will be $\leq$ the number of managed objects in the received alarm cluster. For each of the $S_i, i = 1, \ldots, s$ we calculate the information cost of the set, $I[S_i]$. As a consequence of the independence assumption the information cost $I[S_i]$ is equal to the sum of information costs of the managed objects that belong to the set S_i. In order to find the best explanation of the received alarm cluster we must follow a number of steps (Appendix A). The first step is to find all classes of sets $S_1, S_2, S_3, \ldots, S_l, l \in [1, \ldots, s]$ such that all the received alarms are explained by a fault in S_1 and a fault in S_2 ... and a fault in S_l.

Suppose that we can form n_c such classes. Each of these classes corresponds to a possible solution $R_j, j = 1, \ldots, n_c$. For each of the possible solutions we calculate the information cost of the solution, $I_j = I[R_j]$, $j = 1, \ldots, n_c$. Then we select the solution with the minimum cost. We are interested in investigating how the minimum information cost is changed when we assume that the estimation of the weights for the devices is not correct. In that case we introduce an error $e_i^{'}$ for each of the weights w_i, namely the estimation error. We let I^* be the minimum cost when all the device weights are correct, and I^*_{new} the new minimum information cost when each device weight has an estimation error. We would like to find a bound for the difference $\mid I^* - I^*_{new} \mid$.
Let us denote $I_j^{'}$ as the information cost for the solution R_j when the estimation of the weights is not correct. The new information cost $I_j^{'}$ is related to I_j as follows:

$$\begin{aligned}
\text{Overestimated probabilities:} \quad I_j^{'} &= I_j - \sum_{r=1}^{l_j} e_r^{'}, j = 1, \ldots, n_c & (11)\\
\text{Underestimated probabilities:} \quad I_j^{'} &= I_j + \sum_{r=1}^{l_j} e_r^{'}, j = 1, \ldots, n_c & (12)
\end{aligned}$$

where l_j is the number of devices in the solution R_j, $j = 1, \ldots, n_c$.
If we assume no estimation errors then the selected solution (optimal solution) corresponds to a solution $R_p, p \in [1, \ldots, n_c]$ which has the minimum information cost among all the solutions $R_j, j = 1, \ldots, n_c$. The selected solution has information cost $I_p = min(I_j, j = 1, \ldots, n_c)$. When we introduce the estimation errors the new minimum cost solution will be R_q with information costs $I_q^{'} = min(I_j^{'}, j = 1, \ldots, n_c)$. If we assume overestimated probabilities then the new minimum cost solution R_q will have information cost $I^*_{new} = I_q^{'} \leq I^*$. Also $I_q^{'}$ and $I_p^{'}$ are related as follows:

$$I_q^{'} = min(I_j^{'}, j = 1, \ldots, s) \Rightarrow I_q^{'} \leq I_p^{'} \overset{(11)}{\Rightarrow} I_q - \sum_{r=1}^{l_q} e_r^{'} \leq I_p - \sum_{r=1}^{l_p} e_r^{'} \tag{13}$$

Furthermore the difference between I^*_{new} and I^* ($I^*_{new} \leq I^*$) can be bounded by:

$$\begin{aligned}
& I^*_{new} = I_q^{'} = I_q - \sum_{r=1}^{l_q} e_r^{'} \geq I_p - \sum_{r=1}^{l_q} e_r^{'} = I^* - \sum_{r=1}^{l_q} e_r^{'} \\
\Rightarrow \quad & I^* - I^*_{new} \leq \sum_{r=1}^{l_q} e_r^{'} \leq l_q \max(e_j^{'}, j = 1, \ldots, n) & (14)
\end{aligned}$$

$$\text{Also}\qquad \begin{aligned} I^*_{new} &= I'_q = I_q - \sum_{r=1}^{l_q} e'_r \overset{13}{\leq} I_p - \sum_{r=1}^{l_p} e'_r \\ &\Rightarrow I^*_{new} \leq I^* - \sum_{r=1}^{l_p} e'_r \\ &\Rightarrow I^* - I^*_{new} \geq \sum_{r=1}^{l_p} e'_r \geq l_p \min(e'_j, j = 1, \ldots, n) \end{aligned} \tag{15}$$

By combining (14) and (15) we have:

$$l_p \min(e'_j, j = 1, \ldots, n) \leq I^* - I^*_{new} \leq l_q \max(e'_j, j = 1, \ldots, n) \tag{16}$$

Equivalently if we assume underestimated probabilities then the new minimum cost solution R_q will have information cost $I^*_{new} = I'_q \geq I^*$. Also I'_q and I'_p are related as follows:

$$\begin{aligned} & I'_q = min(I'_j, j = 1, \ldots, s) \Rightarrow I'_q \leq I'_p \\ \overset{(\ 12)}{\Rightarrow} \quad & I_q + \sum_{r=1}^{l_q} e'_r \leq I_p + \sum_{r=1}^{l_p} e'_r \end{aligned} \tag{17}$$

Furthermore the difference between I^*_{new} and I^* ($I^*_{new} \geq I^*$) can be bounded by:

$$\begin{aligned} & I^*_{new} = I'_q = I_q + \sum_{r=1}^{l_q} e'_r \geq I_p + \sum_{r=1}^{l_q} e'_r = I^* + \sum_{r=1}^{l_q} e'_r \\ \Rightarrow \quad & I^*_{new} - I^* \leq \sum_{r=1}^{l_q} e'_r \leq l_q \max(e'_j, j = 1, \ldots, n) \end{aligned} \tag{18}$$

$$\text{Also}\qquad \begin{aligned} I^*_{new} &= I'_q = I_q + \sum_{r=1}^{l_q} e'_r \overset{13}{\leq} I_p + \sum_{r=1}^{l_p} e'_r \\ &\Rightarrow I^*_{new} \leq I^* + \sum_{r=1}^{l_p} e'_r \\ &\Rightarrow I^*_{new} - I^* \geq \sum_{r=1}^{l_p} e'_r \geq l_p \min(e'_j, j = 1, \ldots, n) \end{aligned} \tag{19}$$

By combining (18) and (19) we have:

$$l_p \min(e_j^{'}, j = 1, \ldots, n) \leq I_{new}^* - I^* \leq l_q \max(e_j^{'}, j = 1, \ldots, n) \tag{20}$$

Finally from (16) and (20) we have:

$$l_p \min(e_j^{'}, j = 1, \ldots, n) \leq \mid I_{new}^* - I^* \mid \leq l_q \max(e_j^{'}, j = 1, \ldots, n) \tag{21}$$

Note that we expect that the selected solution will include, most of the time, a small number of managed objects so that $l_q, l_p << n$, where n is the number of managed objects in the alarm cluster. On the average the $\mid I_{new}^* - I^* \mid$ will be bounded by:

$$l_{aver} \min(e_j^{'}, j = 1, \ldots, n) \leq \mid I_{new}^* - I^* \mid \leq l_{aver} \max(e_j^{'}, j = 1, \ldots, n) \tag{22}$$

Where l_{aver} is the expected number of managed objects in a selected solution, and again $l_{aver} << n$.

4.2.2 Accuracy of the Distributed Localization Approach

We return now to our initial problem which is to find a bound for $\mid I^* - I_{dis} \mid$. Suppose that we receive a number of alarms all belonging to the same cluster and m of these alarms cross the boundary between the two domains. If we use the centralized localization approach we can find the optimum solution with information cost I^*. The set of nodes in the solution can be partitioned into two subsets one containing managed objects only in domain one and the other containing objects in domain two. The information costs of these two sets will be I_1^* and I_2^* respectively. Also $I^* = I_1^* + I_2^*$. Now suppose that we use the distributed approach to localize the faults in the same network setting. The manager of domain one will use the probabilistic algorithm in an extended domain which includes the managed objects that are entire in its domain, plus m *proxy* nodes one for each alarm that crosses the boundary between the two domains. Every *proxy* node has an estimated probability of failure assigned to it with an associated estimation error. We assume that the probabilities of failure for the "regular" managed objects in the domain are exact. The information cost of the solution that manager one calculates is I_1. Because of the errors in the estimation of the probability of failure for the proxy nodes I_1 might differ from I_1^*. By using (20) we can bound the difference between I_1^* and

I_1. Therefore $me_{1min} \leq\mid I_1^* - I_1 \mid\leq me_{1max}$, where e_{1min} is the minimum and e_{1max} is the maximum of the estimation errors of the proxy nodes that the manager of domain one *"sees"*. Similarly we can find for domain two that $me_{2min} \mid I_2^* - I_2 \mid\leq me_{2max}$. Also the global solution that the distributed localization approach gives has information cost $I_{dis} = I_1 + I_2$. If we apply the results from the previous section we have:

$$\text{Overestimated Probabilities:} \qquad \left.\begin{array}{l} I_1 \geq I_1^* \\ I_2 \geq I_2^* \\ I_{dis} \geq I^* \end{array}\right\} \Rightarrow$$

$$\Rightarrow \qquad \left.\begin{array}{l} m\ e_{1min} \leq I_1 - I_1^* \leq m\ e_{1max} \\ m\ e_{2min} \leq I_2 - I_2^* \leq m\ e_{2max} \end{array}\right\} \Rightarrow$$

$$\begin{aligned} &\Rightarrow \quad m\ (e_{1min} + e_{2min}) \leq (I_1 - I_1^*) + (I_2 - I_2^*) \leq m\ (e_{1max} + e_{2max}) \\ &\Rightarrow \quad m\ (e_{1min} + e_{2min}) \leq I_{dis} - I^* \leq m\ (e_{1max} + e_{2max}) \\ &\Rightarrow \quad 2m\ \min(e_{1min}, e_{2min}) \leq I_{dis} - I^* \leq 2m\ \max(e_{1max}, e_{2max}) \end{aligned} \tag{23}$$

$$\text{Underestimated Probabilities:} \qquad \left.\begin{array}{l} I_1 \leq I_1^* \\ I_2 \leq I_2^* \\ I_{dis} \leq I^* \end{array}\right\} \Rightarrow$$

$$\Rightarrow \qquad \left.\begin{array}{l} me_{1min} \leq I_1^* - I_1 \leq me_{1max} \\ me_{2min} \leq I_2^* - I_2 \leq me_{2max} \end{array}\right\} \Rightarrow$$

$$\begin{aligned} &\Rightarrow \quad m(e_{1min} + e_{2min}) \leq (I_1^* - I_1) + (I_2^* - I_2) \leq m(e_{1max} + e_{2max}) \\ &\Rightarrow \quad m(e_{1min} + e_{2min}) \leq I^* - I_{dis} \leq m(e_{1max} + e_{2max}) \\ &\Rightarrow \quad 2m \min(e_{1min}, e_{2min}) \leq I^* - I_{dis} \leq 2m \max(e_{1max}, e_{2max}) \end{aligned} \tag{24}$$

By combining (23) and (24) it is easy to show that the difference $\mid I^* - I_{dis} \mid$ is bounded by:

$$2m \min(e_{1min}, e_{2min}) \leq\mid I^* - I_{dis} \mid\leq 2m \max(e_{1max}, e_{2max}) \tag{25}$$

Hence, the difference between the information costs in the two approaches depends on the number of alarms m that cross the boundary between domains. Finally, as it was shown in [6] the estimation error for the weights of a proxy node is expected to be small. Hence, for small values of m , which is the expected case, the difference $\mid I^* - I_{dis} \mid$ will be also small.

5 Conclusions

In this paper we compare the performance between a number of fault localization approaches suitable for a distributed fault management environment. The three proposed methods are namely the *Centralized*, *Decentralized* and *Distributed* Fault Localization approaches. As measures of comparison we used the *accuracy* of the solution and the *complexity* of the identification process that each approach employs. Our comparison proved that the decentralized approach generally has considerably less complexity than the centralized approach, and can provide the same or better solution accuracy. Also the distributed localization approach was proved to have the least complexity of all three schemes in all network settings, but it can not always guarantee an optimum solution. However, as was shown in the previous section, it provides a solution which is almost as accurate as the solution provided by the other two approaches.

References

[1] A.Bouloutas, S. Calo, and A. Finkel. Alarm Correlation and Fault Identification in Communication Networks. Technical Report 17967, IBM T.J. Watson Center, P.O BOX 704, Yorktown Heights, NY 10598, July 1992.

[2] A. Bouloutas, A. Calo, A. Finkel, and I. Katzela. Distributed Fault Identification in Communication Networks. Technical report, IBM T.J. Watson Center, P.O BOX 704, Yorktown Heights, NY 10598, 1993.

[3] A. Bouloutas, G. Hart, and M. Schwartz. Fault Identification Using Finite State Maschine Model with Unreliable Partially Observed Data Sequence. *IEEE Transactions on Communications*, 41:1074–1083, 1993.

[4] Robert H. Deng, Aurel Lazar, and Weiguo Wang. A Probabilistic Approach to Fault Diagnosis in Linear Lightwave networks. *IEEE JSAC*, 11:1438–1449, 1993.

[5] J.F.Jordan and M.E.Paterok. Event Correlation in Heterogeneous Networks Using the OSI Management Framework. unpublished.

[6] I. Katzela, A. Bouloutas, and S. Calo. Comparison of Distributed Fault Identification Methods in Communication Networks. Technical report, IBM T.J. Watson Center, P.O BOX 704, Yorktown Heights, NY 10598, 1993.

[7] Marc Reise. Diagnosis of Communication Systems: Dealing with Incompleteness and Uncertainty. *Qualitative Reasoning and Naive Physics*, pages 1480 – 1485, -.

[8] Marc Reise. Model-Based Diagnosis of Networks: Problem Characterization and survey. *OEGAI-91 Workshop on Model Based Reasoning*, Viennai, 1991.

[9] N. Shroff and M. Schwartz. Fault Detection/Identification in the linear lightwave networks. Technical Report CU/CTR/TR-243-91-24, Center of Telecommunications Research, Columbia University, Center for Telecommunication Research New York, NY 10027, 1989.

[10] Clark Wang and Mischa Schwartz. Identification of Faulty Links in Dynamic Routed Networks. 1992.

[11] Clark Wang and Mischa Schwartz. Identification of Faulty Links in Dynamic-Routed Networks. *IEEE JSAC*, 11:1449–1460, 1993.

A Probabilistic Algorithm for Centralized Identification.

We assume that when the system has at most k faults, where k is a parameter of the algorithm, the algorithm will give the optimum solution. If there are more than k faults in the system then the algorithm will fail to return a solution. The probability that the algorithm will fail to return a solution $1-q$ is equal to the probability that there are more than k faults in the observed system. Hence:

$$\begin{aligned} P_r(\text{algorithm fails}) &= P_r(> k \text{ faults in the network}) \\ &= Pr(> k) \leq 1 - \sum_{i=0}^{k} b(i; N, p) \end{aligned} \tag{26}$$

where N is the number of managed objects that are associated with the received cluster of alarms, p is the $\max(p_j, j = 1, ..., N)$ with p_j the probability of failure for the managed object d_j and $b(i; N, p)$ is the probability of N Bernoulli trials, with probability p of success, and i successes. The equality in (1) holds when all the managed objects in the cluster have the same probability of failure p. For simplicity for now on we assume that all the managed objects in the network have the same probability of failure p.

Description

Input

- A set of nodes N representing the independent managed objects that might fail in a communication network.

- For each managed object the probability of failure and the corresponding information cost $(-logp)$
- A cluster of alarms A. For each alarm we assume that its domain is defined.
- A number k, indicating the maximum number of faults that might have caused the alarm cluster.

Output

- The set of nodes with the minimum information cost that can explain the cluster of received alarms.

Method:
The algorithm runs in k phases:

- In the first phase it searches for a single fault which can explain all the alarms and has the minimum information cost.
- The second phase searches for two faults that could have produced the observed alarms.
- The k^{th} phase searches for k faults that explain the alarms and have the minimum cost.
- Output the solution among the k possible that has the minimum cost.

In the k^{th} phase there are $\binom{N}{k}$ possible combinations. This number is bounded by $O(N^k)$. Note that we have assumed that N is much larger than k.

B Approximating approach for calculating k_1

Case One: $\lceil (N_1+1)p \rceil < \lceil k\frac{N_1}{N} \rceil$

If $\lceil (N_1+1)p \rceil < \lceil k\frac{N_1}{N} \rceil$ then (because $k_1 > \lceil k\frac{N_1}{N} \rceil$) we have also that $k_1 > \lceil (N_1+1)p \rceil$. In this case $b(k_1; N_1, p) > b(k_1+i; N_1, p)$, for $i = 1, \ldots, N_1 - k_1$. Also $b(k_1; N_1, p)$ and $b(k_1+i; N_1, p)$ are related as follows:

$$\begin{aligned} b(k_1+i; N_1, p) &= b(k_1; N_1, p)(\frac{p}{1-p})^i \prod_{j=1}^{i} \frac{N_1 - k_1 - j + 1}{k_1 + j} \\ &< b(k_1; N_1, p)(\frac{p(N_1-k_1)}{(1-p)(k_1+1)})^i, i = 1, \ldots, N_1 - k_1 \end{aligned} \tag{27}$$

The inequality in (27) holds beacause $\frac{N_1-k_1}{k_1+1} > \frac{N_1-k_1-i+1}{k_1+i}$, for $i = 1, \ldots, N_1 - k_1$. Let us denote $\alpha(k_1) = \frac{p(N_1-k_1)}{(1-p)(k_1+1)}$. It is easy to show that $\alpha(k_1) < 1$ for $i = 1, \ldots, N_1 - k_1$. The function $\alpha(k_1)$ decreases with k_1. The same is true for $\frac{1}{1-\alpha(k_1)}$. It is now easy to derive an upper bound for the left side sum in (7):

$$\begin{aligned} &\overset{(\ 27)}{\Rightarrow} \sum_{i=k_1+1}^{N_1} b(i; N_1, p) \le b(k_1; N_1, p) \sum_{i=k_1+1}^{N_1} \alpha(k_1) < \\ &< \quad b(k_1; N_1, p)\frac{1}{1-\alpha(k_1)} < b(k_1; N_1, p)\frac{1}{1-\alpha((N_1+1)p)} = \\ &= \quad b(k_1; N_1, p)((N_1+1)p+1)(1-p) = \binom{N_1}{k_1}(\frac{p}{1-p})^{k_1}(1-p)^{N_1+1}((N_1+1)p+1) < \\ &< \quad \binom{N_1}{N_1/2}(\frac{p}{1-p})^{k_1}(1-p)^{N_1+1}((N_1+1)p+1) \end{aligned} \tag{28}$$

Let us denote $\beta = \frac{p}{1-p}$, $\eta = ((N_1+1)p+1)$ and $\Theta = \binom{N_1}{N_1/2}(1-p)^{N_1+1}\eta$. Then (28) becomes:

$$\begin{aligned} &\overset{(\ 7),(\ 28)}{\Rightarrow} \quad \Theta\beta^{k_1} < \frac{N_1}{N}\sum_{i=k+1}^{N} b(i; N, p) \Rightarrow \\ &\Rightarrow \quad \beta^{k_1} < \frac{\frac{N_1}{N}\sum_{i=k+1}^{N} b(i; N, p)}{\Theta} \Rightarrow \\ &\Rightarrow \quad k_1 > \max\left[\lceil \log_\beta \left(\frac{\frac{N_1}{N}\sum_{i=k+1}^{N} b(i; N, p)}{\Theta}\right)\rceil, \lceil k\frac{N_1}{N}\rceil\right] \end{aligned} \tag{29}$$

Case Two: $\lceil (N_1+1)p \rceil > \lceil k\frac{N_1}{N} \rceil$

If $\lceil (N_1+1)p \rceil > \lceil k\frac{N_1}{N} \rceil$ then we expect that $k_1 < \lceil (N_1+1)p \rceil$. We can then distinguish two cases:

First if N_1 is large then $\sum_{\lceil (N_1+1)p \rceil + 1}^{N_1} b(i; N_1, p)$ is a very small number and can be omitted with no significant error. Hence, (7) becomes :

$$\begin{aligned}
&\stackrel{(7)}{\Rightarrow} \sum_{i=k_1+1}^{N_1} b(i; N_1, p) = \sum_{i=k_1+1}^{\lceil (N_1+1)p \rceil} b(i; N_1, p) + \sum_{\lceil (N_1+1)p \rceil + 1}^{N_1} b(i; N_1, p) \simeq \\
&\simeq \sum_{i=k_1+1}^{\lceil (N_1+1)p \rceil} b(i; N_1, p) < \frac{N_1}{N} \sum_{i=k+1}^{N} b(i; N, p) \Rightarrow \\
&\Rightarrow (\lceil (N_1+1)p \rceil - k_1) b(\rceil (N_1+1)p \lceil; N_1, p) < \frac{N_1}{N} \sum_{i=k+1}^{N} b(i; N, p) \Rightarrow \\
&\Rightarrow (((N_1+1)p+1) - k_1) b(\rceil (N_1+1)p \lceil; N_1, p) < \frac{N_1}{N} \sum_{i=k+1}^{N} b(i; N, p) \Rightarrow
\end{aligned}$$

$$\Rightarrow k_1 > \max\left[\lceil \eta - \frac{\frac{N_1}{N}\sum_{i=k+1}^{N} b(i; N, p)}{b(\lceil (N_1+1)p \rceil; N_1, p)} \rceil, \lceil k\frac{N_1}{N} \rceil\right] \tag{30}$$

Finally if N_1 is small then $\sum_{\lceil (N_1+1)p \rceil + 1}^{N_1} b(i; N_1, p) < b(\lceil (N_1+1)p \rceil; N_1, p)((N_1+1)p+1)$ (as in case one) and (7) becomes:

$$\begin{aligned}
&\stackrel{(7)}{\Rightarrow} \sum_{i=k_1+1}^{N_1} b(i; N_1, p) = \sum_{i=k_1+1}^{\lceil (N_1+1)p \rceil} b(i; N_1, p) + \sum_{\lceil (N_1+1)p \rceil + 1}^{N_1} b(i; N_1, p) < \\
&< \sum_{i=k_1+1}^{\lceil (N_1+1)p \rceil} b(i; N_1, p) + b(\lceil (N_1+1)p \rceil; N_1, p)((N_1+1)p+1) < \frac{N_1}{N} \sum_{i=k+1}^{N} b(i; N, p) \Rightarrow \\
&\Rightarrow (2((N_1+1)p+1) - k_1) b(\lceil (N_1+1)p \rceil; N_1, p) < \frac{N_1}{N} \sum_{i=k+1}^{N} b(i; N, p) \Rightarrow
\end{aligned}$$

$$\Rightarrow k_1 > \max\left[\lceil 2\eta - \frac{\frac{N_1}{N}\sum_{i=k+1}^{N} b(i; N, p)}{b(\lceil (N_1+1)p \rceil; N_1, p)} \rceil, \lceil k\frac{N_1}{N} \rceil\right] \tag{31}$$

C Generalization for multiple domains

So far we compared the different proposed fault localization approaches in the case of two domains. In this section we generalize the analysis in the case of more than two domains. Assume that the network consists of K nonoverlapping domains. Also assume that we would like to compare the centralized and decentralized fault identification approaches in this network setting. Similar to the analysis in section 4.1 we would like to compare the complexity of the two approaches subject to the constraint that both approaches have the same probability of outputing a solution and the decentralized approach must localize at least the same number of faults as the centralized one. In other words we can generalize (3) in the case of multiple domains as follows:

$$\begin{aligned} P_r(\text{Decentralized approach fails}) &\leq P_r(\text{Centralized approach fails}) \\ \text{and} \sum_{i=1}^{K} k_i &\geq k \end{aligned} \tag{32}$$

where k is the number of faults that the centralized approach can localize. In order to compare the complexity of the two approaches we must first find the unknowns k_i in (32). In order to calculate the $k_i s$ we follow the steps of the approximating approach one in section 4.1. We assume that the N managed objects which are associated with the received alarm cluster are partitioned into N_1 in domain one, ..., N_i in domain i, ..., N_K in domain K so that $\sum_{i=1}^{K} N_i = N$. We also assume that the number of faults k_i that each domain manager should localize is proportional to the number of managed objects in the alarm cluster that belongs to the domain. Hence:

$$k_i \geq \lceil k \frac{N_i}{N} \rceil, \quad i = 1, \ldots, K \tag{33}$$

With the proposed approximation we decompose the original complex problem in (32) into K simpler problems one for each domain. Now it is sufficient to solve the problem only for one of the domains. The new problem for domain i can be stated as follows: *Given a probability of failure for the decentralized approach what is the value of the parameter k_i that the domain manager i should use,in the application of the probabilistic algorithm, so that the following two inequalities hold?*

$$\begin{aligned} P_r(> k_i \text{ faults in domain one}) &\leq \frac{N_i}{N} P_r(\text{Centralized approach fails}) \\ k_i &\geq \lceil k \frac{N_i}{N} \rceil \end{aligned} \tag{34}$$

If we follow the same steps as in the approximating approach one in section 4.1, we can calculate k_i. A generalization of Table 1, in the case of multiple domains, is given below:

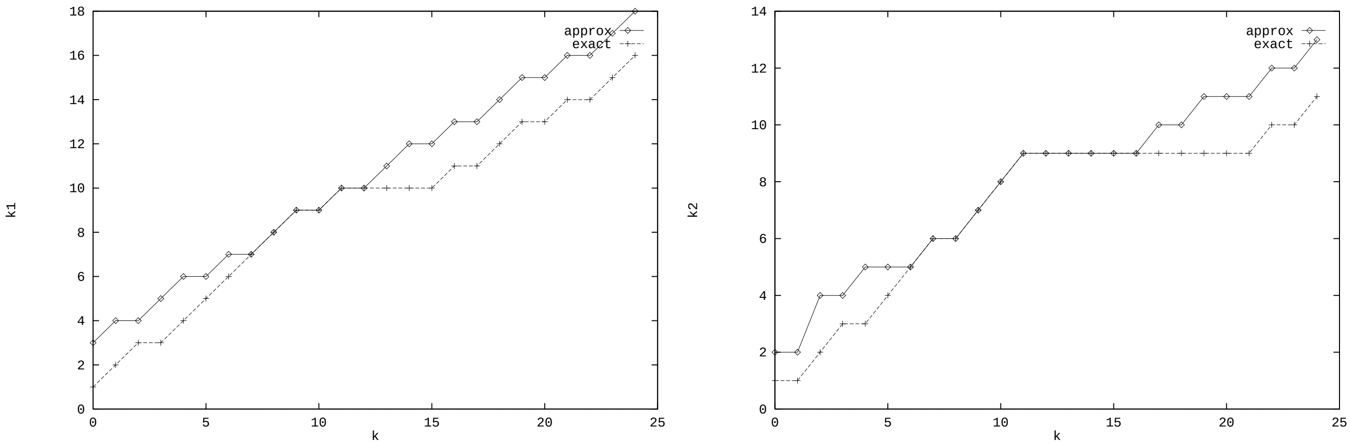
$\lceil (N_i+1)p \rceil < \lceil k\frac{N_i}{N} \rceil$	$k_1 > \max\left[\lceil \log_\beta \left(\frac{\frac{N_i}{N}\sum_{i=k+1}^{N} b(i;N,p)}{\Theta_i} \right) \rceil, \lceil k\frac{N_i}{N} \rceil \right]$
$\lceil (N_i+1)p \rceil > \lceil k\frac{N_i}{N} \rceil$, N_i large	$k_1 > \max\left[\lceil \eta_i - \frac{\frac{N_i}{N}\sum_{i=k+1}^{N} b(i;N,p)}{b(\lceil (N_i+1)p \rceil;N_i,p)} \rceil, \lceil k\frac{N_i}{N} \rceil \right]$
$\lceil (N_i+1)p \rceil > \lceil k\frac{N_i}{N} \rceil$, N_i small	$k_1 > \max\left[\lceil 2\eta_i - \frac{\frac{N_i}{N}\sum_{i=k+1}^{N} b(i;N,p)}{b((N_i+1)p;N_i,p)} \rceil, \lceil k\frac{N_i}{N} \rceil \right]$
$\beta = \frac{p}{1-p}$, $\eta_i = (N_i+1)p+1$	$\Theta_i = \binom{N_i}{N_i/2}(1-p)^{N_i+1}\eta_i$

Figure 2: Formulas for estimating the number of faults that domain manager i must calculate.

Finally the average complexity of the decentralized approach in the case of more than two domains will be (generalization of (10)):

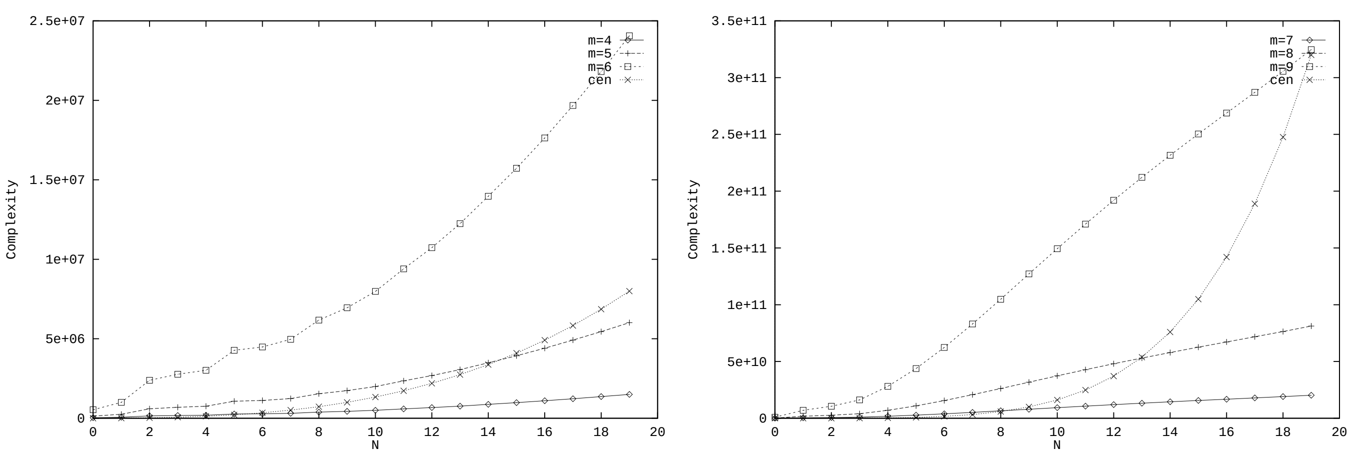
$$
\begin{aligned}
C_{dec}^{aver} &= \max_{1\le j\le K}(2^{m_j})\sum_{i=1}^{N}\frac{N!}{i!(N-i)!}\max\left(i^{k_{i1}},(N-i)^{k_{i2}}\right) = \\
&= \max_{1\le j\le K}(2^{m_j})\sum_{i=0}^{\lceil \frac{N}{2} \rceil}\frac{N!}{i!(N-i)!}i^{k_i}
\end{aligned} \tag{35}
$$

Where m_j is the number of alarms, in the received alarm cluster, that cross the boundary between domain j and any other of the $K-1$ domains. Specifically $m_j = \sum_{\substack{i=1 \\ i\neq j}}^{K} m_{ji}$ where m_{ji} is the number of alarms that cross the boundary between domain j and i.



(a) N =100, p = 0.01 (b) N = 100, p = 0.01

Figure 3: k_1, k_2 versus k



(a) k=3, p = 0.1 (b) k=5, p = 0.1

Figure 4: Complexity versus N

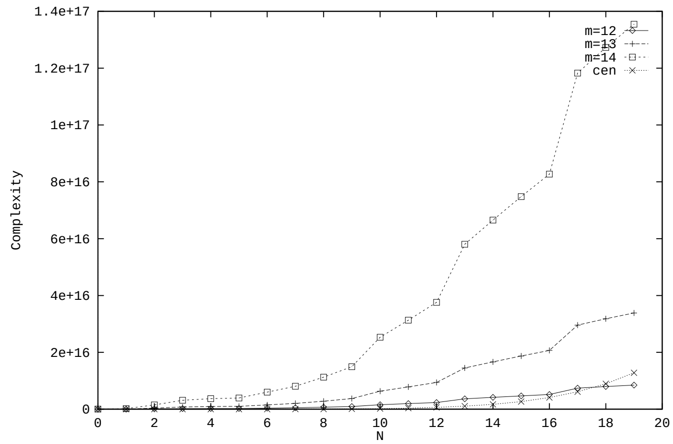
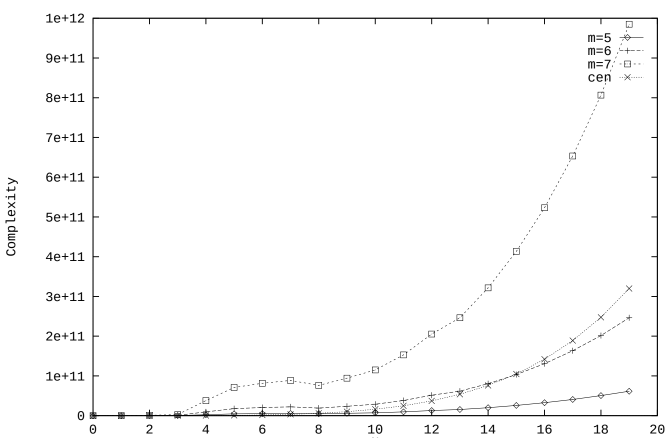
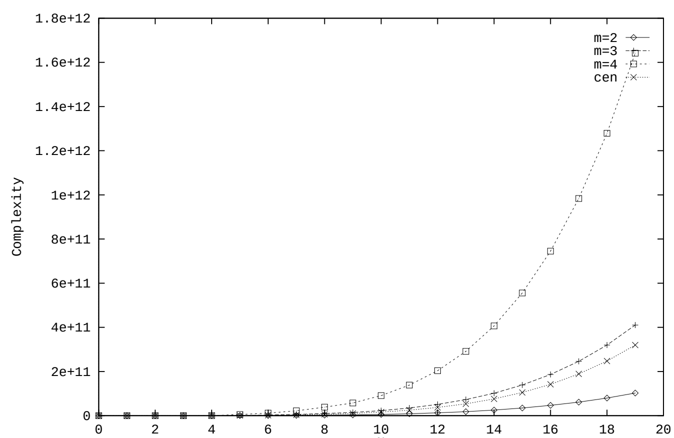


Figure 5: Complexity versus N, p=0.1, k=7



(a) k=5, p = 0.01



(b) k=5, p = 0.001

Figure 6: Complexity versus N