Alarm Correlation and Fault Identification in Communication Networks

A. T. Bouloutas, S. Calo, and A. Finkel

Abstract—We present an approach for modeling and solving the problem of fault identification and alarm correlation in large communication networks. A single fault in a large network may result in a large number of alarms, and it is often very difficult to isolate the true cause of the fault. This appears to be one of the most important difficulties in managing faults in today’s networks. The problem may become worse in the case of multiple faults. In this paper we present a general methodology for solving the alarm correlation and fault identification problem. We propose a new alarm structure, propose a general model for representing the network, and give two algorithms which can solve the alarm correlation and fault identification problem in the presence of multiple faults. These algorithms differ in the degree of accuracy achieved in identifying the fault, and in the degree of complexity required for implementation.

Keywords—network management, alarm correlation, fault identification, fault management, fault localization

I. INTRODUCTION

Communication networks have increased dramatically in size and complexity in the last few years. A typical network may consist of hundreds of nodes from various manufacturers with different traffic and bandwidth requirements. The increasing complexity poses serious questions of network management and control. One aspect of network management is fault management, and a central component of fault management is fault identification. Even though failures in large communication networks are unavoidable, quick detection and identification of the cause of failure can make communication systems more robust, and their operation more reliable, ultimately increasing the level of confidence in the services they provide.

The process of fault identification in large communication systems can be divided into three steps:

- The first step is fault detection. Fault detection can be thought of as an on-line indication that some component of the network is malfunctioning. Usually communication network devices provide indications (in the form of alarms) when they sense a malfunction. Thus, fault detection can be accomplished using the alarms provided by the network devices.

- The second step is fault localization, i.e., the analysis of the alarms from the devices of the network in order to propose possible hypotheses of faults. This step is essential since, in most cases, alarms do not provide explicit or detailed identification of the malfunctioning device.

- The third step is fault identification. The actual fault is isolated given a number of possible hypotheses of faults. Testing may be the most appropriate way of isolating the fault at this stage.

This paper concentrates on the second step of the above mentioned process. A framework is developed for analyzing the information within the alarms emitted by communication network devices in order to propose possible hypotheses of faults. The proposed framework can be also used to perform alarm correlation, i.e., to reduce the number of alarms and messages presented to the network operators. This appears to be an important problem in the operation of today’s networks; the network operators are overwhelmed with messages and the fault localization task is made very difficult. Too much information has the same effect as too little information—fault identification is made more complex.

In section 2 we propose a representation for communication systems for the purpose of fault identification, in section 3 we present the nature of the problem, in section 4 we study the information carried by the alarms, in section 5 we propose algorithms and methods for solving the alarm correlation problem, in section 6 we give a general framework for identifying multiple faults, and in section 7 we present two examples.

II. SYSTEM REPRESENTATION

Since we are interested in identifying faults in communication systems, we need to represent faults. Before representing faults we need to define what we mean by the term fault. Here, fault identification means identifying the device that is at fault and the nature of the fault.

If we assume that each device has a single fault mode (can only fail in one way), then fault identification means the identification of the device at fault. If a device can fail in more than one way, then fault identification means the identification of the device at fault and the nature of the fault.

From this discussion it is clear that the representation of the faults of any system is closely related to the representation of the structure of the system. In the rest of this section we focus on the representation of the structure of communication systems, i.e., the representation of the faults for a single fault mode per device. This representation is then extended to include the nature of the fault, i.e., the same framework can be used to represent the faults for the case of multiple fault modes per device.

The authors are with the IBM Research Division, T.J. Watson Research Center, Yorktown Heights, N.Y. 10598
IEEE Log Number 9400935.

0090-6778/94$04.00 © 1994 IEEE

0090-6778/94$04.00 © 1994 IEEE
Communication systems consist of processes, which consist of subprocesses, down to the level of hardware or software components that can be considered indivisible (the appropriate level of division is system and user dependent). A natural candidate (but not the only candidate) for representing communication systems is a Phrase Structured Grammar. ¹ This is so because a phrase structured grammar allows sentences to be formed from expressions which are formed from sub-expressions, etc., thus giving a natural way to represent hierarchically organized complex structures.

A Phrase Structured Grammar (PSG) can be defined as the 4-tuple \( G = (V, T, P, S) \) where \( V \) and \( T \) are finite sets of variables and terminals respectively. It is assumed that \( V \) and \( T \) are disjoint. \( P \) is a finite set of productions and \( S \) a special variable which is designated as the root symbol. Each of the productions in \( P \) are of the form \( A \rightarrow a \) where \( A \in V \) and \( a \in (V \cup T)^* \). The formula \((V \cup T)^*\) denotes the Kleene closure [2] of the set \( V \cup T \). Thus, productions consist of a left hand side which is a variable symbol and a right hand side which consists of any number of variable or terminal symbols. For example the production \( A \rightarrow B \cdot C \cdot D \) means that \( A \) is formed by conjoining clauses of types \( B, C, \) and \( D \) in that order. While \( A \) has to be a variable symbol, symbols \( B, C, \) and \( D \) can be either variable or terminal.

We introduce two new production forms that enable us to abbreviate the set of productions \( P \). If the clauses \( A \rightarrow B, A \rightarrow C \) and \( A \rightarrow D \) belong in the set of productions, then we can replace them with the Alternative Forms Clause \( A \rightarrow B|C|D \). In this clause \( A \) has to be a variable symbol and \( B, C, \) and \( D \) have to be either variable or terminal symbols. This indicates that \( A \) may consist of any of the clauses in the right hand side. Another abbreviation is the use of the superstar notation. The clause \( A \rightarrow B^* \) means that \( A \) can be substituted by any number of \( B \) (including zero). Thus, this clause is equivalent to \( A \rightarrow \lambda | B | B \cdot B | B \cdot B \cdot B | \ldots \), with \( \lambda \) the empty string.

In representing the possible faults of a communication system, we can use the representation of its structure. The terminal symbols of this representation could be the faults we need to identify. Since communication networks are hierarchically organized, we can use this organization to represent our system. The basic logical unit in representing communication networks is the communication connection. The network consists of communication connections. The communication connection provides service to two terminal points, and the combination of the terminal points and the communication connection may constitute the communication connection for two higher level terminal points. Communication fails because either of the terminal points has failed, or the communication connection has failed. For example, assume that a particular network consists of four nodes and three links, link-AB, link-BC, and link-CD. We can represent the network as a collection of links:

\[
\text{NETWORK} \rightarrow \text{LINK-AB} \cdot \text{LINK-BC} \cdot \text{LINK-AB} \cdot \text{LINK-BC} \cdot
\]

Each link can be further expanded into the components that it consists of:

\[
\begin{align*}
\text{LINK-AB} & \rightarrow \text{NODE-A} \cdot \text{CHANNEL-AB} \cdot \text{NODE-B} \\
\text{LINK-BC} & \rightarrow \text{NODE-B} \cdot \text{CHANNEL-BC} \cdot \text{NODE-C} \\
\text{LINK-CD} & \rightarrow \text{NODE-C} \cdot \text{CHANNEL-CD} \cdot \text{NODE-D}
\end{align*}
\]

If there exists a logical link utilizing the services of link-AB and link-BC we can write:

\[
\text{LOGICAL-LINK-AC} \rightarrow \text{LINK-AB} \cdot \text{LINK-BC}
\]

This representation of the system is equivalent to a representation of the possible faults, assuming a single fault mode for each network component. Multiple fault modes can be included by expanding any symbol representing a network component. For example if \( \text{Node-A} \) has two fault modes we can expand it as follows:

\[
\text{NODE-A} \rightarrow \text{NODE-A} \cdot \text{F1} \cdot \text{NODE-A} \cdot \text{F2}
\]

The two fault modes of \( \text{Node-A} \) are considered terminal symbols, and thus cannot be expanded any further.

In a similar fashion we can expand the channels. For example Channel-AB may consist of a connector to Node-A, a line, and a connector to Node-B. The line may consist of an Encryptor, a T-1-Channel, and another Encryptor. The representation could be as follows:

\[
\begin{align*}
\text{CHANNEL-AB} & \rightarrow \text{CONNECTOR-A} \cdot \text{LINE-AB} \cdot \text{CONNECTOR-B} \\
\text{LINE-AB} & \rightarrow \text{ENCRYPTOR-T1} \cdot \text{CHANNEL-ENCRYPTOR} \\
\text{CONNECTOR-FAULTY} & \rightarrow \text{CONNECTOR-ENCRYPTOR} \\
\text{ENCRYPTOR-FAULTY} & \rightarrow \text{ENCRYPTOR-T1} \cdot \text{CHANNEL}
\end{align*}
\]

This technique provides a representation of any component in the network which may fail. Furthermore, this approach may be used to represent both the structure and the faults of a communication network. With this approach, a communication network is represented as an acyclic graph whose terminal symbols describe the devices which could be at fault: the dependence graph. Each node of this graph (representing a device or a communication connection) is expanded into the devices upon which the correct operation of the node depends, or into the fault modes that may appear.

### A. PSGs, Dependence Graphs, and Independent Faults

Our motivation for the introduction of PSGs for the representation of the system has been the attempt to introduce a general approach for representing alarms, faults and system structure. This approach can help the conceptual organization of the work and can guide the design of algorithms. However, PSGs should not be considered as the universal representation of all systems but only as a tool which can give guidance in the design of algorithms.

An alternative representation of the system is the use of a dependence graph. The operation of the system depends on

---

¹Phrase Structured Grammars are no different than Context Free Grammars [2], but here we follow the terminology in [3].

²Transmission line
the operation of subsystems which depend on the operation of basic devices and components.

We can represent systems and devices as nodes in a graph. If the operation of a device depends on the operation of some other device then there is an arc connecting the two devices. If a device fails then by traversing the graph one can find all the devices which could have possibly caused the fault. All the independent devices which could be at fault and could have caused the faults in the system are leaves of the graph. Note that generally the dependence graph is a directed graph and if someone could traverse it starting at any node (the node represents a system), then he would reach a set of nodes which represent the independent devices that could be at fault. It is exactly these devices we are interested in identifying at fault. This model would easily address fault propagation because once an independent device fails faults propagate to all the devices they depend on it.

Example

In Fig. 1 (a) we present an example of a simple three node network.

Nodes $S$, $R_1$, $R_2$ are linked by links $S-R_1$ and $R_1-R_2$. Suppose that $S$ sets up a one-to-two broadcasting $B$ with $R_1$ and $R_2$ by two paths $P_1(S,R_1)$ and $P_2(S,R_2)$. Then the representation of the dependence relations of the system can be seen in fig. 1(b). In terms of a PSG the system can be described as follows:

$$B \rightarrow P_1 \cdot P_2$$

$$P_1 \rightarrow L(S-R_1), \quad P_2 \rightarrow L(S-R_1) \cdot L(R_1-R_2).$$

Obviously, the paths’ $P_1$ and $P_2$ failure probabilities are not independent because $\text{Probability}(P_2 \text{ fail} | P_1 \text{ fail}) = 1$ and this is reflected in the dependence graph in fig. 1(b). Both nodes have at least one common descendant. However, the leaf nodes $S$, $S-R_1$, $R_1$, $R_1-R_2$, $R_2$ are considered to be independent and this is where we would like to identify the fault. If nodes $S$ and $S-R_1$ were not independent then this would mean that there is another object (node).

Both would depend on. This would then be the terminal object and the object we would be interested in identifying at fault.

It is our intention in this paper to only consider independent sources of fault. If devices and systems which are not independent are associated in any way with a possible fault then this fault is projected to the independent devices by following down the dependence graph. An example of this is shown in Fig. 2. A dependence graph and two alarms $A1$ and $A2$ affecting devices $D4$, $D5$, $D6$, $D7$ are presented. Clearly devices $D5$ and $D6$ are not independent since they have a common descendant. However if we project the alarms down the dependence graph we note that alarm $A1$ is equivalent to alarm $A5$ affecting devices $D14$, $D15$, $D16$, $D17$, $D18$. In a similar fashion we can project alarm $A2$ to alarm $A3$. (Note that here we have not used any negative information meaning that we have not used the fact that since $D21$ has not indicated any problem then devices $D14$ and $D15$ can not be at fault. This would limit alarm $A5$ to $A4$ –shown in dashed lines. This is generally examined in section 6.) This way the dependence graph can help us project the possible faults to the independent devices. It is the dependence graph we propose to represent as phrase structure grammars, and clearly this can be easily done.

It may seem that PSGs are more than we need. However, sometimes a dependence graph may not be enough. Consider the case where a channel consists of two subchannels. The channel is operational if any of the subchannels is operational. This is difficult to model using a dependence graph but it is easy to model using a PSG. That is why we chose PSGs as a general representation model which could represent both structure, faults, and alarms. In the rest of the paper we assume that structure is described either by a PSG or a dependence graph which in turn is described by a PSG.
III. Nature of the Problem

In this section we focus on the nature of the problem, and attempt to explain why fault identification and alarm correlation in communication systems are difficult problems. Communication networks consist of devices independently manufactured by various vendors. While the internal implementation of many devices vary from vendor to vendor, the interface of each device with the rest of the network has been standardized to conform to widely accepted standards (e.g., SNA, OSI, etc.).

Each of these devices has been independently designed. The designer of a communication system device has to ensure that both his device and its perceived interface, i.e., the rest of the network projected into the device's observation space, are working correctly. Naturally, the design process includes designing alarms for the various fault conditions that the device may encounter in operation. Thus, the designer needs to provide two kinds of alarms:

- Alarms for faults that exist within the device; and,
- Alarms for faults that appear in the interface with which the device has to conform.

A particular fault in a device may disrupt its operation as well as its behavior towards other devices. This fault may cause many devices to emit alarms indicating problems with their interfaces. In many cases, a network operator may be overwhelmed with different alarms all due to the same network problem. Even though it may appear that more information helps to diagnose the problem, in fact the opposite is too often true. Usually alarm messages do not explicitly carry the information needed to diagnose a fault. Alarms describe in detail the faulty condition, i.e., the symptom of the fault, but do not, in most cases, describe the cause of the fault.

IV. Information Carried by the Alarms

In order to design a fault identification system we need to study the information carried by the alarms. The ideal alarm is presented in [5]. There, the SNA alert structure is introduced. SNA alerts try to answer the following questions about any fault: Who, What, Where, When, and Why. However, because of lack of knowledge sometimes Where and Why are not provided.

Who: The name of the system reporting, or if different the name of the system experiencing the fault.
What: The condition of the fault, i.e., the symptom of the fault.
Where: A description of the position in the network where the problem occurred.
When: A description of the time at which the problem was detected.
Why: The cause of the problem, i.e., the nature of the fault.

If this information were available, then fault identification and alarm correlation would be very easy. In order to identify the fault that triggered an alarm, we would have to check the Where and Why field of the alarm. In order to correlate alarms, we could simply notice that alarms with the same Why, Where, and similar When fields indicate the same problem, and thus should be correlated.

However, the information described above is usually not provided by the alarms of communication systems devices, because each device of a communication network has only limited information about the rest of the system. Most alarms emitted by network devices report the device that is experiencing the fault, the symptoms of the fault, and the time of the detection of the fault, i.e., answer the questions, Who, What, and When. However, for the purpose of fault identification and alarm correlation we need more. We need answers to the questions Where and Why which are usually not provided.

The information added by the Where and Why fields gives us a better ability to distinguish faults and to correlate the alarms attributed to those faults. The trade-off between the amount of the available information and the fault distinguishing ability (alarm correlation ability) is made clear in the following two cases:

- If we make the simplifying assumption that two faults can not happen at the same location at the same time, then, for the purpose of fault identification, we only need the answer to the question Where.
- If we make the simplifying assumption that the whole network (or the portion of the network we are monitoring) experiences only a single fault at a time and alarms appear immediately after the fault, then we could identify faults and correlate alarms using only timing information.

V. Proposed Approach

It is clear, from the discussion above, that alarms that attribute the cause of the fault to the same network device and have similar When fields should be correlated. The problem is that most alarms do not give explicit fault-localization information. Thus, we propose to associate explicit fault-localization information with each alarm.

Most alarms do not include fault-localization information because the location of the fault is not known precisely by the device emitting the alarm. However, almost every alarm contains implicit fault-localization information. For example, consider the alarm "LOS-DTE" emitted by a Channel Service Unit (CSU). This alarm indicates a loss of the signal from the data-terminal-equipment side of the CSU, and not from the side connected to the T-1 transmission line. This alarm, even if it does not indicate a specific location for the fault, restricts the possible locations of the fault to a smaller part of the network. In similar fashion, bipolar violation, or BPV (an alarm emitted by a CSU), is an alarm that indicates a problem with the transmission line and not with the CSU. This alarm restricts the possible

\[\text{correlation}\]
locations of the fault to locations outside the CSU.

Thus, each alarm can be associated with a set of possible locations representing the locations of the fault. Note that we propose to associate each alarm with all the possible locations of the fault and not only with the most probable ones. In the case that alarms are reliable and there is only a single fault in the network, then fault localization is straightforward: The fault lies in the intersection of the set of locations indicated by each alarm. Thus, intuitively, alarms that share a common intersection should be correlated.

In order to formally define the algorithm for alarm correlation (and fault localization) we introduce the term "incident." An incident is a set of correlated alarms. Ideally, an incident should contain all the alarms attributed to the same fault. Any incoming new alarm can cause the creation of a new incident, the association of this alarm with an existing incident, or a totally new reorganization of incidents. This is so because a new alarm may make a new hypothesis of faults more probable.

Incidents may be opened and closed. While the creation of an incident is triggered by the arrival of an alarm, the deletion of an incident is not straightforward. An incident should be deleted (closed) when the faulty condition which caused the generation of the alarms correlated by the incident ceases to exist. This raises an interesting question: How can we verify that a fault that has triggered a set of alarms is resolved? We believe that, in an integrated fault management environment, testing is the most appropriate way to establish that a problem is resolved. Since testing is beyond the scope of this paper, we can assume that closing an incident is beyond the scope of this work. For simplicity, we may assign the responsibility of closing an incident to the network operator. Sometimes a simple timer may be used to close an incident. This is possible if the delay of all the alarms generated due to a failure of a component is much less than the time between failures of that component.

We have to note that we are only considering independent faults. Given the dependence graph that represents the structure of the system—possibly described by a PSG—we assume that the only devices that may fail are the ones that are terminal symbols in the PSG, or leaf nodes in the dependence graph. Alarms appear at any level in the dependence graph but these alarms can be projected to the independent devices by following down the dependence graph. Note that usually we do not have to reach to the leaves of the graph in order to reach devices that are mutually independent and this can speed up the algorithm proposed later in the paper. However, for simplicity and consistency we assume that all alarms are projected to the leaves of the graph as shown in figure 2.

The algorithm for alarm correlation can be formally defined as follows:

A. Positive Information Algorithm (PIA, Part I)

Input: One new incoming alarm, a set of incidents and their associated alarms, and a description of the network topology (the dependence graph defined by a PSG).

Output: A new set of incidents indicating the probable fault locations.

Method:

Step 1 Given a new alarm and a description of the network topology, identify the possible locations of the fault indicated by this alarm by localizing the alarm and projecting the alarm down the dependence graph.

Step 2 Construct the minimum number of incidents such that all the alarms associated with an incident share a common intersection, and all alarms are assigned to some incident.

Step 2 of the algorithm is important since incidents indicate faults. Correct association of alarms with incidents would mean correct fault localization. In step 2 we seek the minimum number of incidents (thus faults) that can "explain" the appearance of a set of alarms. There is an implicit assumption here: It is more likely to have few faults than many. If there is a single fault to be identified, then the number of incidents is one. All alarms should have a common intersection since they are attributed to the same fault. Step 2 of the algorithm will localize the fault to the part of the network defined by the intersection of all the alarms.

A problem arises when there is more than one fault that needs to be identified. In this case, step 2 of the algorithm may have more than one solution. Given a set of alarms, there may be more than one way of constructing the incidents, i.e., more than one way of proposing possible hypotheses of faults. One example highlighting this case is when, given three alarms, we get three intersections of their fault localization areas. Any combination of two intersections would be enough to define the incidents. The choice of the two best is undetermined at this point.

One way of resolving this problem is to associate a-priori probabilities of failure with each network component. We would then choose the incidents that contain components that are more likely to fail.

Instead of associating a probability of failure with each component, we can associate an "information cost," the negative of the logarithm of the probability of failure. Even though information costs are equivalent to probabilities for independent faults, working with information costs instead of probabilities has certain advantages. One of these advantages is the intuitively justified additivity property: the probability that two independent faults \((f_1, f_2)\) appear in the network is \(p_{f_1} \cdot p_{f_2}\), while the corresponding information cost is \(I(p_{f_1}) = I(p_{f_1}) + I(p_{f_2})\).

Using the above method we can refine the second step of the algorithm as follows:

- Associate three pieces of information with each inci-

\[\text{as described elsewhere in the paper we only consider independent sources of fault}\]
dent:
- The "fault," i.e., a single component of the network that may have caused the alarms;
- The information cost associated with the component that may have caused the alarms (corresponding to the a-priori knowledge about the fault); and,
- All the alarms that contain this fault in their fault localization field.

- Find a set of incidents such that the sum of the information costs of the incidents contained within this set is minimized, and all alarms are associated with some incident.

This method produces the most probable faults, that is, the faults that "best explain" the observed alarms.

It can be proven (by transforming this problem to the problem of "Hitting Set" [4]) that the problem of finding the set of incidents that minimize the information cost is NP-Hard. Below, we formally restate the problem and give a heuristic algorithm that approximates the optimal solution (and in many cases finds it).

B. Positive Information Algorithm (PIA, Part II)

Input: A finite set of possible fault locations \( F \), an information cost \( I(f) \in \mathbb{Z}^+ \) for each \( f \in F \), and a set \( A \) of subsets of the set \( F \). \( A \) contains elements \( A_j \subseteq F, j = 1, 2, \ldots n \), where \( n \) is the number of observed alarms. Each \( A_j \) is a set of possible fault locations as described by the fault localization field (see below) of the \( j \)-th alarm.

Output: A subset \( F' \subseteq F \) such that \( I(F') = \min_{G \in G} \sum_{g \in G} I(g) \).

\( G \) is in \( G \) if and only if \( G \subseteq F \) and \( G \cap A_j \neq \emptyset \) for all \( A_j \in A \).

That is, \( F' \) is the set \( G \) which minimizes the information cost and explains all the alarms.

Method:

Step 1 \( F' \leftarrow \emptyset \).

Step 2 Assign in \( F' \) the element \( f \) of \( F \) that is contained in the most sets \( A_j \), where \( A_j \in A \). If there is more than one such element \( f \), choose the one with the least cost.

Step 3 Delete all elements of \( A \) that contain \( f \). \( A \) is thus reduced. If \( A \) is empty, output \( F' \), otherwise go to step 2.

As can be verified, the heuristic method described above can be implemented in polynomial time. As long as the information costs associated with the elements of the set \( F \) do not have large variations, the method should closely approximate the desired minimum.

Even though the approach proposed by PIA is natural, simple, and intuitive, there are various issues that remain to be resolved. One important issue concerns methodologies for associating fault localization information with each alarm. Such a methodology must take into account the diverse topologies of communications networks, as well as the widely differing fault localization information provided by the texts of different alarms.

- Some alarms are sophisticated enough to pinpoint the location of the fault, e.g., faults in the cards of an IDNX\(^{10}\) can be detected by its peer unit.
- Some alarms can distinguish the direction of the fault, e.g., the alarm "LOSS DTE" in a CSU refers to the loss of the signal from the data terminal equipment side of the CSU and not from the T-1 side.
- Some alarms can only distinguish whether a fault that a device is experiencing is external or internal to that device.

One approach is to introduce a set of primitives which describe the possible locations of the fault (internal, external, layer above, layer below, channel, peer unit, etc.), and associate each alarm with one member of this set. In this case, we must construct a function that maps each alarm to a primitive in this set. Ideally, this function should be independent of the network topology, so that changes to the communications network do not require a change in the function. An additional function may be constructed that maps the alarm-primitive pair to specific network devices.

In this paper we assume that the set of possible fault locations indicated by each alarm is contained within the alarm. For clarity, we assume that each alarm has a fault localization field that contains this information. We do not go into implementation details on how this is done, but this is one of the implementation issues that must be considered in any real system.

VI. GENERAL FRAMEWORK USING ALARM VOLATILITY

Even though PIA performs alarm correlation, it may occasionally fail to identify the correct fault. The reason is that the algorithm does not take into account the absence of alarms, and does not provide for unreliable alarms. The absence of alarms may sometimes be a valuable piece of information. An example may help in demonstrating this point.

Assume that we have five pieces of equipment connected in series: \( A \) is connected to \( B \), which is connected to a \( T-1 \) line, which is connected to \( C \), which is connected to \( D \). Thus, the topology is as follows: \( A-B-(T-1)-C-D \). Assume that we get an alarm from \( A \) stating that either \( B \), or \( (T-1) \), or \( C \), or \( D \), or the line between \( A \) and \( B \), or the line between \( C \) and \( D \) is at fault. We then get an alarm from \( D \) stating that either \( A \), or \( B \), or \( (T-1) \), or \( C \), or the line between \( A \) and \( B \), or the line between \( C \) and \( D \) is at fault. These alarms share a common intersection, thus should be correlated. Line \( T-1 \) will be pointed to by PIA as the most probable cause of these alarms. Communication lines are presumably more likely to be at fault than any other device. While these alarms are correlated correctly, the identification of the fault may not be correct. Absence of alarms from \( B \) and \( C \) indicate that the problem is not with the \( T-1 \) line. It is, most probably, associated with a fault in the line that connects \( A \) to \( B \),

\(^{10}\)A high performance intelligent multiplexer
or a fault in the line that connects C to D. This example highlights the fact that the absence of alarms is important information that should be used in the fault identification process.

The other case ignored by PIA is the existence of unreliable alarms. Sometimes alarms are emitted because a threshold was exceeded, or some other transient condition happened, and the alarm is not an indication of a permanent fault. Thus, alarms can be unreliable from the fault identification point of view since an investigation of a non-existent fault may be triggered.

In this section, we introduce a general framework that accounts for both conditions ignored by the Positive Information Algorithm, i.e., the information carried by the absence of alarms, and the existence of unreliable alarms.

It would be useful at this point to contrast our work with the approach presented in [6,7]. Even though the authors present there a probabilistic theory of diagnosis given a set of malfunctions (manifestations) they do not address the problem of unreliable manifestations (alarms in our case.) One of their assumptions is the “Mandatory Causation Assumption” which states that no effect can occur without being caused by some disorder. Our approach is more general and more suited to fault identification in our particular area of application, namely communication networks.

Since alarms are now considered unreliable, the problem becomes a different one. We no longer want to find only the faults, but we want to jointly estimate the faults and the actual alarms. Thus, we would like to estimate the sentence $S(A,F)$; here $F$ represents the faults and $A$ the alarms. This is the sentence that describes the faults and the actual alarms. The sentence $S(A,F)$ can be expanded to $S(F)$, a sentence that describes the faults, and to $S(A|F)$, a sentence that describes the alarms given the faults:

$$S(A,F) \rightarrow S(A|F) \cdot S(F)$$

This rule actually represents a family of rules that differ only in $F$. This gives rise to an estimation problem where the trade-off between the order of the model, or structural complexity (number of faults in our case), is weighed against the fit to the data (the observed alarms in our case). An excellent study on this topic can be found in [3]. We will modify the framework proposed in [3] to the particular needs of our problem.

The sentence that describes the faults can be expanded to any number of faults that are chosen from a finite set of faults:

$$S(F) \rightarrow \text{fault}^*$$

$$\text{fault} \rightarrow f_1 \mid f_2 \mid \ldots \mid f_n$$

Here $f_1, f_2, \ldots, f_n$ are the terminal symbols of the grammar that represents the structure and the faults. A fault should appear at most once in the sentence $S(F)$.

Next we need to expand the sentence that represents the alarms observed given the faults:

$$S(A|F) \rightarrow S(A_+|F) \cdot S(\text{DELETE}|A_+) \cdot S(\text{ADD})$$

$$A = (A_+ - S(\text{DELETE}|A_+)) \cup S(\text{ADD})$$

This sentence is expanded to a sentence that represents the a-priori knowledge of the alarms given the faults—the expected alarms $A_+$, a sentence that corresponds to addition of alarms, and a sentence that corresponds to deletion of alarms. This sentence essentially represents the correct alarms and the noise. The last condition (Equation (5)) needs to be added for correctness. It states that the observed alarms should be the ones expected minus the ones deleted plus the ones added.

It may seem that the sentence $S(A_+|F)$ carries a-priori knowledge that is difficult to get in a real network. Indeed it may be difficult to find all the alarms a set of faults can produce. These alarms may depend on the network topology, the time the fault occurred, the state of the network at the time of the fault. In order to avoid this problem we can use information abstraction. We can represent the alarms given a fault to any degree of detail we want. We may only represent the source of the alarm (the device that emitted the alarm), or we may represent both the source and the information contained within the alarm, or we may even choose to expand $S(A_+|F)$ to any set of alarms, indicating no a-priori knowledge about the alarms given the faults.

Variable degrees of information abstraction can be included in the same framework. We do not have to use the same degree of detail of the alarms given the faults for all faults. For some faults it may be easy to find the alarms associated with them, and for others it may be very difficult. The framework proposed so far can include both of these cases.

The sentence $S(\text{ADD})$, can be expanded to any number of alarms, among the alarms that can be generated, and the sentence $S(\text{DELETE}|A_+)$ can be expanded to any number of alarms among the alarms that are included in $A_+$. Thus we can write:

$$S(\text{ADD}) \rightarrow \text{Alarm}^*$$

$$\text{Alarm} \rightarrow A_1 \mid A_2 \mid \ldots \mid A_n$$

$$S(\text{DELETE}|A_+) \rightarrow \text{Alarm}^*$$

$$\text{Alarm} \rightarrow A_1^* \mid A_2^* \mid \ldots \mid A_n^*$$

Here, $A_1^*, A_2^*, \ldots, A_n^*$ are alarms that belong to set $A_+$. One important characteristic of the class of problems we are considering is the multiplicity of solutions. Usually, alarms do not describe in detail the fault that caused their generation. A single alarm may be due to many faults; a set of alarms could be due to a number of possible faults.

The multiplicity of solutions also becomes apparent when we examine the case of multiple faults. Usually, there are many combinations of faults that could have caused the observed alarms; finding all of them could be very difficult and non-informative. It is clear that we need a way to choose the “best” solution among a set of possible ones. One way to do this is to use weights. Following the proposed framework, solutions (or faults) should be weighted using two kinds of weights:
A weight based on the a-priori knowledge about the probability that a specific set of faults appeared; and,

- A weight based on the knowledge about the probability that the set of faults proposed has caused the set of observed symptoms (alarms).

Thus, in estimating the sentence that jointly describes the alarms and the faults we need a way of weighting faults and alarms. The minimum information framework introduced by Hart [3] is applicable here. In [3], information costs are associated with the sentences of phrased structured grammars. The sentence which describes the observation and the system and is associated with the minimum information, is proposed as the best estimate of the system.

In [3], two expansion rules are presented for associating information costs to phrase structured grammars:

- If \( U \) is expanded as \( U \rightarrow U_1 \cdot U_2 \cdots U_n \), then
  \[
  I(U) = I(U_1) + I(U_2) + \cdots + I(U_n).
  \]

- If \( U \) is expanded as: \( U \rightarrow U_1 | U_2 | \cdots | U_n \), then
  \[
  I(U) = I(U_1) + I_1 \text{ when } U \text{ is rewritten as } U_1. \text{ This is equal to the information cost associated with } U_1, \text{ plus } I_1. \text{ } I_1 \text{ is the information associated with the choice of } U_1 \text{ among the various possibilities. For our purposes } I_1 = -\log p_{U_1} \text{ where } p_{U_1} \text{ is the probability of choosing } U_1 \text{ among the various other possibilities. In case we cannot associate probability measure to the various choices, then } I_1 = \log n. \text{ This way we give equal probability to any choice. A thorough investigation of the various ways of assigning information measures to phrase structured grammars is given in [3].}

Now consider a specific sentence that describes a number of faults where \( F \) is given by:

\[
S(F) \Rightarrow f_1 \cdot f_2 \cdots \cdot f_k. \tag{10}
\]

Where the symbol \( \Rightarrow \) is used to indicate an actual derivation using the production rules. The expansion rules described above lead to the equation:

\[
I(S(F)) = I(f_1) + I_1 + I(f_2) + I_2 \cdots + I(f_k) + I_k \tag{11}
\]

Here \( I_1 = -\log p_{f_1} \) and \( p_{f_1} \) is the probability that the fault \( f_1 \) appears. The information cost associated with \( f_1 \) is \( I(f_1) \). Following [3], we will henceforth assume, without loss of generality that \( I(f_1) \) is zero.

In order to associate information with \( S(A|F) \) the first expansion rule may be used to expand \( I(S(A|F)) \) as follows:

\[
I(S(A|F)) = I(S(A_1|F)) + I(\text{ADD}) + \cdots + I(S(\text{DELETE}|A_1)) \tag{12}
\]

Given \( F \), the information cost \( I(S(A_1|F)) \) is constant, since we can uniquely describe the alarms given the faults (a-priori knowledge). The information associated with the random addition of alarms is similar in structure to the information associated with the faults. If:

\[
S(\text{ADD}) \Rightarrow A_1 \cdot A_2 \cdots A_k
\]

then the expansion rule leads to:

\[
I(S(\text{ADD})) = I(A_1) + I(A_2) + I(A_3) \cdots (14)
\]

\[
\cdots I(A_k) + I_{A_k}
\]

Here \( I_{A_k} = -\log p_{A_k} \) and \( p_{A_k} \) is the probability that alarm \( A_k \) was accidentally emitted. \( I(A_k) \) is the information associated with alarm \( A_k \). Since alarm \( A_k \) is not further expanded this information cost is zero, [3]. The sentence \( S(\text{DELETE}|A_k) \), which describes the accidental deletion of alarms, can be expanded in similar fashion.

Once we have defined the information costs for the various parts of equation (1), the problem of fault identification can be defined as an optimization problem. The objective is to find a set of faults, that is a function of the observed alarms, such that the information cost associated with the joint description of the alarms and the faults, is minimized, i.e.,

\[
\min_{F \in \mathcal{F}} I(S(A|F)) = \min_{F \in \mathcal{F}} \{ I(S(A|F)) + I(S(F)) \} \tag{15}
\]

where \( \mathcal{F} \) is the set of all potential faults in the network. Equation (15) is a general description of the family of problems presented so far. For example, appropriate cost assignments could transform the equation into the problem addressed by the Positive Information Algorithm described in section 5 provided:

- Given a fault \( f \), \( S(A_1|f) \) contains only alarms which have \( f \) in their fault localization field. We may assume \( I(S(A_1|f)) \) is zero.

- The alarms are reliable. Therefore the probability of alarms being accidentally deleted or emitted is zero. The information associated with the term \( S(\text{ADD}) \) and the term \( S(\text{DELETE}|A_k) \) is infinite. Consequently, the minimization problem in (15) is restricted to fault sets \( F \) where the expected set of alarms from the faults in \( F \) is given by \( A \), the observed set of alarms.

Equation (15), in general, presents a hard discrete optimization problem. Only heuristic algorithms can be considered since the problem can be proven to be NP-Hard. The general idea of any heuristic algorithm that is trying to find a solution for equation (15) is that the algorithm has to search the space of the candidate faults, and the space of the possible alarms in order to locate the ones that minimize equation (15). The best and the fastest heuristic algorithms can be proposed only if we take into account the particular cost function assigned to the faults and the alarms of the network we are managing.

We present here a general algorithm that should be considered more as a guideline for designing an algorithm to suit a particular network.
A. Positive and Negative Information Algorithm (PNIA, Part I)

Input: A communication network and a set of alarms. The communication network has the following a-priori information associated with it:

- The set of possible faults in the network (a representation of the network),
- The set of possible alarms,
- Fault localization information associated with each of the alarms,
- Information about the possible alarms associated with each fault,
- Information cost associated with each fault, and
- Information cost associated with accidental addition and deletion of any alarm.

Output: The set of incidents indicating the most probable faults and the most probable alarms in the network.

Method: This method constructs a tree; each node of the tree represents a possible solution. Specifically, each node in the tree is represented by a pair, \((F, A)\). Here \(F\) represents some set of faults and \(A \subseteq A_o\) where \(A_o\) is the set of observed alarms. The space of all such possible points can be rather large. This method finds the best solution among the nodes examined.

Assume that any observable alarm can be explained by at least one fault. The root node of the tree is given by \((\emptyset, A_o)\). A leaf node in the tree is given by \((F_l, \emptyset)\). Let \((F_p, A_p)\) represent a node on the tree, where \(A_p \neq \emptyset\). Child nodes of this node are constructed as follows:

Step 1: Find the faults (i.e., terminal symbols of the grammar representing the structure) that are contained in the fault localization field of the largest number of alarms in the set \(A_p\).

Step 2: For each fault \(f\) identified in step 1 create a child node of \((F_p, A_p)\). The child node consists of the point \((F_c, A_c)\) where \(F_c = F_p \cup \{f\}\) and \(A_c\) is obtained by deleting from \(A_p\) all the alarms that contain \(f\) in their fault localization fields. Naturally, \(A_c \subseteq A_p\), and \(F_p \subseteq F_c\).

Each node in the tree represents a possible combination of faults which could have produced the observed alarms \(A_o\). A node \((F_p, A_p)\) in this tree has an associated information cost given by:

\[
I(S(A_p, F_p)) = I(S(A_p | F_p)) + I(S(F_p)) = (16) \\
I(S(A_o | F_p)) + I(S(ADD)) + I(S(DELETE | A_p)) + I(S(F_p))
\]

Here \(A_o\) represents the set of expected alarms given the fault set \(F_p\). The most probable set of faults corresponds to the node with the minimum information cost associated with it. In practice this node is typically a leaf node. Note that at a leaf node \(S(ADD) = \emptyset\).

Although equation (15) is NP-Hard, the tree constructed in PNIA Part I is usually quite manageable. Suppose \((F_p, A_p)\) is a node in the tree and \(f \in F_p\). The condition that \(f\) be contained in the fault localization field of some alarm in \(A_o\) markedly reduces the search space. An alarm typically has only a few faults in its fault localization field. Furthermore, the depth of the tree is no greater than the cardinality of \(A_o\). In practice it is often not necessary to expand the entire tree, as low cost paths often glaringly suggest themselves. We formalise our approach of looking for minimal cost nodes with the:

B. Positive and Negative Information Algorithm (PNIA, Part II)

Method:

Step 1: Given a communication network and a set of alarms, apply PNIA Part I, and construct the tree representing the possible combinations of faults.

Step 2: Find the node (does not have to be a leaf) with the minimum information cost.

Step 3: Output the incident that corresponds to faults in the path from the root to the node with the minimum information cost.

One of the steps of this algorithm requires the information associated with the observed alarms given the faults. This is not difficult to estimate since we know \(S(A_o | F)\) and the observed alarms. We have to find the additions and deletions of alarms with the minimum information such that the expected alarms (given by \(S(A_o | F)\)) are transformed to observed alarms. In order to do that we need to make the assumption that the sequence of the observed alarms is not important. This is a natural assumption since there are random delays involved in observing the alarms. If the above mentioned assumption is true, then, in order to get the set of observed alarms by transforming the set of expected alarms, we need to add and delete alarms. The set of alarms that should be added or deleted can be found by taking the difference of the two sets: the set of observed minus the set of expected alarms is the set of alarms that should be added; and, the set of expected alarms, minus the set of observed alarms is the set that should be deleted.

VII. Examples

A. Example 1

We now introduce a simple example in order to clarify the proposed algorithm. Assume that we have a communication system that consists of six components that may fail: \(f_1, \ldots, f_6\). We have assumed a single fault mode per component. The information costs associated with these faults are: \(I_{f_1} = 3, I_{f_2} = 5, I_{f_3} = 7, I_{f_4} = 3, I_{f_5} = 5, I_{f_6} = 1\). Assume that we observe three alarms: \(A_{123}\). The fault localization fields of those alarms are as follows:

- \(A_1 = \{f_1, f_2, f_3\}\), \(A_2 = \{f_3, f_4, f_5\}\), and \(A_3 = \{f_5, f_6, f_1\}\).
- For simplicity we will assume that the alarms of this system are reliable. Thus, the sentence that describes the alarms given the faults is a sentence that describes the observed alarms. We also assume that the information cost associated with an alarm given a fault is infinite if the fault is not in the alarm's fault localization field, and zero otherwise.
That is, given a fault a particular alarm will either always appear or it will never appear.

In applying PIA Part I, in step 1, we find the faults that are contained in most of the alarms. These are faults $f_1, f_3,$ and $f_5$. Each of these faults is contained in two alarms. We can expand each of the faults.

If we choose $f_1$ as the fault associated with the first potential incident, the cost of $f_1$ is 3 and alarms $A_1$ and $A_3$ are associated with this incident.

In expanding $f_1$ we delete $A_1$ and $A_3$ and repeat the first step. If we repeat the first step (now we are left only with alarm $A_2$) we have three potential faults to expand: $f_3, f_4$, and $f_5$. Among them the one with the least cost (most probable) that can explain the existence of alarm $A_2$ is $f_4$. Thus this solution gives two possible faults, $f_1$ and $f_4$.

After expanding fault $f_1$ we can expand faults $f_3$ and $f_5$. These give us two more hypotheses of possible faults. If fault $f_3$ happened, then, most probably, fault $f_6$ also happened with total cost 8. If fault $f_5$ happened, then, most probably, fault $f_7$ happened, with total cost 8. Among the examined hypotheses of faults, the one with cost 6 (faults $f_1$ and $f_4$) is the most likely one.

From the algorithm described above we can see the way that the tree of the possible solutions is generated. The root is connected to three nodes, $f_1, f_3$, and $f_5$. Each of these nodes can be expanded to a set of leaves. Node $f_1$ is expanded into three leaves, $f_3, f_4$, and $f_5$. Each of these leaves represents two faults: fault $f_1$ and one of the faults $f_3, f_4$, or $f_5$. In a similar fashion node $f_3$ can be expanded into leaves $f_1, f_6$, and $f_5$. Finally, node $f_5$ can be expanded into leaves $f_3, f_2$, and $f_1$.

Since we assume that the alarms received are reliable, the information associated with the intermediate internal nodes $f_1, f_3,$ and $f_5$ is infinite. Take the case of $f_1$; the rest of the cases are similar. Suppose $f_1$ is the only fault in the network. Alarms $A_1$ and $A_2$ contain fault $f_1$ in their fault localization fields. Alarm $A_2$ is not explained by fault $f_1$, and must be a random event. Thus, alarm $A_2$ must be deleted. But, according to our assumptions, the information cost associated with the deletion of an alarm is infinite. Thus the information cost associated with the internal node $f_1$ is infinite, and $f_1$ alone cannot explain the observed alarms.

B. Example 2

Consider a simple communications network consisting of five pieces of equipment connected in series: $A$ is connected to $B$, which is connected to a $T-1$ line, which is connected to $C$, which is connected to $D$. Symbolically the network topology is as follows: $A - B - (T-1) - C - D$. Assume the $T-1$ line and each device in the network have only a single fault mode and that each device can emit only a single alarm. Assume also that the lines $l_{AB}$ between $A$ and $B$; and $l_{CD}$ between $C$ and $D$ also have a single fault mode.

The expected alarm for fault $f_A$ is $a_A$, and $I_{fa} = 4$. The expected alarm for fault $f_B$ is $a_B$, and $I_{fb} = 4$. The expected alarm for fault $f_C$ is $a_C$, and $I_{fc} = 4$. The expected alarm for fault $f_D$ is $a_D$, and $I_{fd} = 4$. The expected alarms for fault $f_{T-1}$ are $a_A, a_B, a_C$ and $a_D$ and $I_{f_{T-1}} = 1$. The expected alarms for $f_{AB}$ are $a_A$ and $a_D$. The information cost of this fault is 2. The expected alarms for $f_{CD}$ are $a_A$ and $a_D$. The information cost of this fault is 3. Assume that the cost of addition or deletion of any alarm is 3.

Finally, assume $I(S(A_e|F)) = 0$ for any fault set $F$. The cost of any particular fault $f$ is given by:

$$I(S(ADD_{f}))+ I(S(DELETE_{f})) + I(f) \quad (17)$$

Suppose that two alarms are observed: alarm $a_A$ whose fault localization field indicates that either $B$, or $(T-1)$, or $C$, or $D$, or $l_{AB}$ or $l_{CD}$ is at fault; alarm $a_D$ whose fault localization field indicates that either $A$, or $B$, or $(T-1)$, or $C$, or $l_{AB}$ or $l_{CD}$ is at fault.

The $T-1$ line has the lowest cost associated with it and may at first glance appear to be responsible for the two alarms. However, the total cost given by (5) associated with a fault in the $T-1$ line is 7 since alarms $a_B$ and $a_C$ are expected but not observed. The cost of deletion of each of these alarms is 3. The cost of a fault in the line $l_{AB}$ however is 2. It can easily be verified that this possibility is in fact the solution of minimum cost. The observed alarms are due to a fault in the connection between $A$ and $B$.

Although the $T - 1$ device is the most likely device to fail in the network, it is not always at fault. In this case the absence of alarms from $B$ and $D$ is valuable information that points us towards a different fault. This example points out the importance of comparing observed alarms with expected alarms. As can be verified, the low cost solution given by PIA Part II is $T-1$.

VIII. Conclusions

The advent of large scale heterogeneous wide area networks and the recent proliferation of local area networks make alarm correlation an important consideration for communication network management systems. We have presented a general framework for modeling and solving the problem of alarm correlation and fault identification in communication networks.

Instead of undertaking a rule based approach in which patterns of alarms are indications of specific faults, we associate fault localization information with each alarm. Given that each alarm contains such information, general algorithmic schemes have been developed in order to localize a fault and correlate alarms in a more efficient and extendable fashion.

The framework developed for fault localization has been extended to cover multiple faults in a more realistic environment where even the observed alarms can be unreliable. The problem then becomes a discrete optimization problem where the objective is to find the set of faults and the set of alarms that minimize a certain cost function. Since this problem is NP-Hard, heuristic algorithms have been proposed for its solution.
Although the proposed approach is a general framework for designing fault identification systems, more research needs to be continued in many areas. The algorithms presented here rely heavily on a-priori information which is either guessed or can be experimentally gained. One key research direction could be the way to adaptively acquire the complete information by starting with an empty, an incomplete subset, or even an incorrect set of information.

REFERENCES


Anastasios Bouloutas was born in Lamia, Greece in 1969. He received his Ph.D. in 1989 from the Department of Electrical Engineering of Columbia University. For the year 1990 he worked as a Postdoctoral fellow at IBM Research. From 1991 to Sept. of 1992 he worked as a research scientist at the National Technical University of Athens Greece and he participated in RACS and ESPRIT projects. Since Sept. 1992 he has been a Research Staff Member at IBM Research. His research interests include systems and network management.

Dr. Saphir B. Calo received his Ph. D. degree in Electrical Engineering from Princeton University in 1976. Since 1977 he has been a Research Staff Member in Computer Sciences at the IBM T. J. Watson Research Center, Yorktown Heights, New York, and has worked and published in the areas of queuing theory, data communication networks, multi-access protocols, expert systems, and complex systems management. He has managed research projects in the communications and systems performance areas and has served on the staff of the IBM Research VP, Systems. He joined the Systems Analysis Department in 1987, and is currently Manager of Systems Applications. This research group is involved in studies of architectural issues in the design of complex software systems, and the application of advanced technologies to systems management problems. As part of his interest in systems management research, Dr. Calo was instrumental in establishing the IEEE International Workshop on Systems Management, and served as its first Chairman. Dr. Calo holds two United States patents.

Dr. Allan Finkel is a Research Staff Member and project leader of the “Systems Management” project of the IBM Research Division. Dr. Finkel received a bachelor’s degree from the State University of New York at Binghamton in 1977 and a Ph. D. in mathematics in 1982 from New York University where he studied at the Courant Institute of Mathematical Sciences. During 1983-1983 he was a Member of the Institute for Advanced Study at Princeton, New Jersey. He joined the Mathematical Sciences Department of IBM Research in the fall of 1983 as a postdoctoral fellow and moved to the Computer Sciences Department in 1985. His research interests include expert systems, systems and network management and object-oriented databases.