J. Indian Inst. Sci., Nov.–Dec. 1996, 76, 827–847.

Realistic abductive reasoning-based fault and performance management in communication networks

G. PREM KUMAR* AND P. VENKATARAM
Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560 012, India.
(Tel: (+91) (080) 3340855; Fax: (+91) (080) 3347991, e-mail: {prem, pallapa}@ece.iisc.ernet.in)

Received on December 16, 1994; Revised on July 27, 1995.

Abstract

Abductive reasoning is identified as a suitable candidate for solving network fault and performance management problems. A method to solve the *network fault diagnosis problem* using *realistic abductive reasoning model* is proposed. The realistic abductive inference mechanism is based on the parsimonious covering theory with some new features added to the abductive reasoning model. The network diagnostic knowledge is assumed to be represented in the most general form of causal chaining, namely, hyper-bipartite network. As many explanations may still be generated by the realistic abductive reasoning model, we propose a probabilistic method to order them so as to try out the diagnostic explanation in the decreasing order of plausibility until the *hard failure*-like faulty device is isolated and replaced/corrected.

In contrast, *performance degradation* in communication networks can be viewed to be caused by a set of faults, called *soft failures*, owing to which the network resources like bandwidth cannot be utilized to the expected level. An automated solution to the performance management problem involves identifying these soft failures and use/suggest suitable remedies to tune the network for better performance. Abductive reasoning model is used again to identify the network soft failures and suggest remedies. Common channel signalling network fault management and Ethernet performance management are taken up as case studies. The results obtained by the proposed approach are encouraging.

Keywords: Network fault diagnosis, network performance management, realistic abductive reasoning model, parsimonious covering theory, common channel signalling network fault management, Ethernet performance management.

1. Introduction

As the networks are growing geographically and the number of heterogeneous devices supported by them is increasing exponentially, the management of such networks plays a vital role[1]. Network fault and performance management are very important and complex issues of present-day network management. Expert system technology has been widely used to solve the network fault diagnostic problem[2–5]. Based on the observed symptoms, a diagnostic expert system attempts to isolate the faults and recommend remedial action. On the other hand, the performance management is in its infancy and much of the work is yet to get into it to meet expected performance in the network. In this paper, we first discuss the network fault management and later show that network performance management can be considered as a special case of network fault management where the network performance degradation is viewed to be caused by a set of soft failures.

* Present address: Motorola India Electronics Ltd, The Senate, 33A, Ulsoor Road, Bangalore 560 042, India. email: prem@miel.mot.com

There are specialized problems that have to be addressed by network fault diagnosis systems. The entire diagnostic information may not be available at once and there may be missing information. In both the cases, the management centre needs to confirm with the respective managed nodes before initiating the diagnostic process.

The fundamental idea behind abductive reasoning is "reasoning to the best explanation"[6]. Based on the given symptoms, initially, it uses forward chaining to anticipate all the possible disorders, and then it uses backward chaining to confirm if the explanation is supported to a required degree of confidence. Ever since parsimonious covering theory[7–10] was developed for abductive reasoning with sound mathematical foundation, there has been a shift in attention from deductive to abductive reasoning. Abductive reasoning generates all the possible explanations which may require further refinement to arrive at appropriate explanations[11]. Deductive reasoning, though generates only appropriate explanations, will not generate those required explanations which it would, if the missing information were to be present. Both the abductive and the deductive reasoning strategies are far from reality to use in network management applications. Hence, in the proposed model for network fault and performance management, we use the realistic abductive reasoning model (or Realistic_ARM)[12] to solve the problem. The Realistic_ARM is a compromise between the two strategies and attempts to find very appropriate explanations for a given set of symptoms. In this model, the diagnostic knowledge is represented in the most general form of causal chaining, namely, hyper-bipartite network. The proposed probabilistic extension to the realistic abductive reasoning model orders the obtained explanations for the network fault isolation and correction so that a more plausible explanation can be tried out before a less plausible explanation.

We present in Section 2 the notation used and the realistic abductive reasoning model in Section 3. Section 4 describes the complexity of the network fault diagnostic problem and a restricted common channel signalling network fault knowledge model to illustrate how the realistic abductive reasoning model solves the problem. In Section 5, solution to the network performance problem is proposed and Ethernet performance management is discussed. Conclusion follows in Section 6.

2. Notation

Although abductive reasoning models are based on simple causal networks, they provide theoretical foundation for a variety of real-world applications.

Definition 1: The *diagnostic problem*, P, is a 4-tuple $< M,D,H,L >$ where $M = \{m_1, m_2, ..., m_e\}$ is a set of manifestations causing a set of disorders, $D = \{d_1, d_2, ..., d_f\}$ either directly or *via* a set of hypotheses (a hypothesis could be a manifestation or a disorder), $H = \{h_1, h_2, ..., h_r\}$. And, $L = \{l_{ij} | i \in M \cup H;\ j \in H \cup D\}$ is a set of causal links joining any two related elements in M, H and D. In a general case, there are many causes to each of the manifestations, many effects to each of the disorders, and both causes and effects to each of the hypotheses.

Definitions 2: *Hyper-bipartite network* is an acyclic graph, $G = < M,D,H,L >$, where M is a set of manifestations (in the bottom-most layer), D, a set of disorders (in the top-most layer) and H, a set of hypotheses (in one or more intermediate layers). All elements of M, H, and D are represented as nodes in their respective layers. And, L is a set of edges joining any two related nodes in M, H and D. Let the number of layers in the graph be N, denoted by $M, P, Q, \ldots, Z, D$.

Definition 3: *Layered network* is an acyclic graph $G^* = \langle M, D, H^*, L^* \rangle$, constructed from the hyper-bipartite network G, where each node belonging to M, H^* and D are connected only to the nodes in its neighbouring layers. The procedure to convert a hyper-biparite network into a layered network, *Build_Layered_Net*, is discussed in Appendix I.

Definition 4: A *symptom* is an observed manifestation/hypothesis/disorder.

Definition 5: A *volunteered symptom* is a hypothesis/disorder at layer i $(1 < i \leq N)$ observed to be present.

A hypothesis/disorder *covers* a symptom if there is a causal pathway from the hypothesis/disorder to the symptom.

Definition 6: *A cover* or an *explanation* is a set of hypotheses/disorders that covers all the given symptoms.

In solving the diagnostic problem, P, where the representation is in the form of a layered network, G^*, *jth* cover of layer i $(1 \leq i < N)$, $c_j^i = \{h_1, h_2, \ldots, h_s\}$ is a set of disorders at layer $(i + 1)$, which covers the symptoms at layer i. At each layer, there may be more than one explanation for the given symptoms and they are placed in the *cover-set* of that layer, $C_i = \{c_1^i, c_2^i, \ldots, c_t^i\}$. While at the top-most layer, a volunteered symptom is simply added to each cover of the cover-set if it is not already present.

Definition 7: *Intermediate cover* (t_k^i), of layer i, is a cover belonging to the cover-set (T_i) being generated, which provides an explanation for the symptoms being explored but may or may not provide explanation for the unexplored symptoms.

Definition 8: *Direct disorder*, $dd \in D$, of a manifestation/hypothesis is the direct cause of the manifestation/hypothesis mapping on to the top-most layer.

Definition 9: *Irredundancy* is the parsimonious criteria used in Realistic_ARM to refine the cover-set by eliminating the redundant covers. A cover c_j^i is redundant if there exists another cover c_k^i, which is a subset of c_j^i.

Definition 10: The *probability of disorder* is the probability with which the disorder d_k occurs. This is an expert-assigned probability, denoted by $P(d_k)$.

The hypotheses at the intermediate layers are not assigned any probabilities like $P(d_k)$ by the expert. Instead, based on the symptoms observed at lower layers, we assign the plausibilities of the cover as the probability of hypothesis at next layer if the hypothesis exists in the cover. If it exists in several covers of the lower layer, the maximum of such plausibilities is assigned as the probability with which the hypothesis exists. If a manifestation/hypothesis is observed as a symptom, that is given a probability of 1.0 (*i.e.*, the manifestation/hypothesis is fully confirmed as a symptom).

Definition 11: *Probability the manifestation m_j is caused by the disorder d_k* is the expert assigned probability with which the manifestation occurs once the disorder exists, denoted as $P(m_j/d_k)$ (or p_{kj}).

Definition 12: For each of the explanations in layer A, a plausibility $P\left(c_n^A, M^+, M^-\right)$ will be assigned which represents the likelihood of the explanation for a given set of symptoms M^+ that are found to be present and a set of symptoms M^- that are found to be absent.

Definition 13: *Solution* to a diagnostic problem is the set of all explanations for the given symptoms.

3. The realistic abductive reasoning model

Realistic abductive reasoning model[12] is a modified version of the abductive reasoning model[7] to solve the diagnostic problems effectively in a realistic scenario. This model uses abductive inference mechanism based on the parsimonious covering theory with some new features added to the general model of diagnostic problem solving.

The inference process used in abductive reasoning that is based on parsimonious covering theory is similar to the model of sequential *hypothesis-test* cycle of human diagnostic problem solving[10]. The 'hypothesis' part of it is *covering* the given symptoms and *hypothesis updation* to obtain *parsimonious* covers. The 'test' part of it is the question–answering process to explore more symptoms for *hypothesis discrimination*. This cycle continues, taking one symptom at a time, until all relevant questions are asked and all symptoms are processed.

The diagnostic knowledge in Realistic_ARM is represented in the form of a hyper-bipartite network. In this model, all the manifestations/ hypotheses have direct disorders. All the elements belonging to M,D,H^* exist only in their respective layers. Any symptom belonging to any layer may appear at any time during the reasoning process. All the possible manifestations that *could* be present in a layer because of the existing manifestations through common disorders (common disorder is a hypothesis/disorder, a manifestation causes along with some other manifestations/hypotheses) are queried at once before starting the reasoning process for that layer. The advantage of querying for all the possible manifestations at once is two fold: (i) all the covers will be generated with the same set of symptoms, and (ii) especially in the networking environment, queries for the presence of manifestations need a lot of time in collecting the information and it is good to present them at the earliest.

Solution to the diagnostic problem where the knowledge base is represented in the form of a hyper-bipartite network is found by converting it into a layered network and solving it as a series of bipartite networks, moving upwards one layer at a time. The algorithm and other related subroutines are presented in Appendix I.

The elements of a cover (say, jth cover) for the symptoms in layer (i–1), c_j^{i-1} become symptoms for layer i. C_0 is initialized to $\{\varnothing\}$. In addition to these, some more symptoms that are added at layer i by user input or interactive querying, Q_j^i, together form jth *symptom-set* for layer i, denoted by s_j^i.

$$s_j^i = c_j^{i-1} \cup Q_j^i.$$

An intermediate cover-set T_i corresponding to s_j^i is built as follows:

(i) For the first symptom of the symptom-set, all its causes form different intermediate covers since each of them separately provides an explanation for that symptom.

(ii) For each of the subsequent symptoms:

(a) if an intermediate cover provides explanation for the symptom, it will remain unchanged;

(b) otherwise, for an intermediate cover, t_k^i, append only those causes of the symptom, m_l, which are supported by 'prespecified number of symptoms', one at a time to form new intermediate covers and delete t_k^i; and

(c) if no new intermediate cover is generated, then append direct disorder of the symptom to the intermediate cover.

After the covers are built to provide explanation to all the symptoms of the symptom-set, the probability assignment to each of the covers in T_i, is done as follows.

For each of the explanations E in T_i, three measures are computed. *Measure* 1, denoted as $M1(E, M^+)$, is the likelihood based on the symptoms M^+ that are present. *Measure* 2, denoted as $M2(E, M^-)$, is the likelihood based on the manifestations/hypotheses M^- that are supposed to exist when the cover E is concluded but are found to be absent. The fault diagnosis is viewed as *closed problem solving*, *i.e.*, if a symptom is not found, it is considered to be absent. *Measure* 3, denoted as $M3(E)$, is the likelihood of the cover E, based on expert-assigned probability of the disorder. $M3$ is calculated only at the topmost layer and it is assumed to be 1.0 at the intermediate hypotheses.

Definition 14: The relative likelihood of an explanation, E, in any layer for given M^+ and M^- is given by $M(E, M^+, M^-) = M1 * M2 * M3$, where

$$M1\left(E, M^+\right) = \prod_{[m_j \in M^+]} P\left(m_j\right) \cdot \left(1 - \prod_{[d_k \in E]} \left(1 - p_{kj}\right)\right).$$

$$M2\left(E, M^-\right) = \prod_{[d_k \in E]} \prod_{[m_l \in M^-]} \left(1 - p_{kl}\right).$$

$$M3(E) = \prod_{[d_k \in E]} \frac{P\left(d_k\right)}{1 - P\left(d_k\right)}.$$

$M1$ is a result of the generalized Bernoulli formula for independent events of the measure $P(m_j)$. $M2$ is the measure obtained by considering the diagnostic problem solving as an instance of closed problem solving from the symptoms which are observed to be absent. $M3$ is the relative likelihood of the cover based on expert-assigned probability of the disorder as discussed above [10].

T_i is then appended to the cover-set C_i and reinitialized to $\{\emptyset\}$ to take up next symptom-set of that layer. When all symptom-sets of the layer are explored, C_i is made irredundant. This process repeats for all the layers till the top-most layer is reached. At the top-most layer, the volunteered symptoms are simply added to each cover of the cover-set if they are not already present. After *covering* all symptoms of the top-most layer, the reasoning process repeats from the bottom-most layer if any more symptoms are left uncovered. The intention here is to *cover* the symptoms only at their respective layer along with other symptoms of that layer to avoid excessive guess in generating the explanations and retain the simple-layered network architecture without additional dummy nodes (for details, refer to Kumar and Venkataram[12]).

3.1. *Properties of Realistic_ARM*

The modifications and the special features incorporated into Realistic_ARM have shown good results over the existing abductive reasoning models, in particular Pure_ARM (we call the most general form of the inference mechanism used by the existing abductive reasoning models as pure_ARM. A brief algorithm for Pure_ARM is given in Appendix II.) We show a couple of results by proving the following theorems.

Theorem 1: The number of covers generated by the Realistic_ARM is always less than the number of covers generated by pure_ARM. (Except for a special case mentioned in Remark 1.)

Proof: Consider a symptom, m_l, belonging to one of the symptom-sets being explored at ith layer. For an intermediate cover, t_k^i, of the cover-set being generated due to this symptom-set, which is present in both Realistic and Pure_ARM, if t_k^i is not able to provide an explanation for m_l, we show that the number of covers added by Realistic_ARM is always less than and subset of that added by Pure_ARM. Note that the intermediate cover-sets, T_i^P and T_i^R, are initially $\{\emptyset\}$ for each set of symptoms, and after all the symptoms of that set are explored they are added to the cover-sets, C_i^P and C_i^R, respectively. The superscripts P and R denote the cover-sets generated by Pure and Realistic_ARM, respectively.

Let n be the number of causes of m_l, including the direct disorder. Using Pure_ARM, all the n causes of m_l may enter T_i^P forming n new intermediate covers (if the set covering principle permits to form new intermediate covers). Whereas using Realistic_ARM in the best case (with respect to minimum number of intermediate covers being added) when no cause of m_l is supported by any other symptom, only the direct disorder enters T_i^R, forming only one intermediate cover. In the worst case (with respect to maximum number of intermediate covers being added), when all the causes other than direct disorder have the support of the prespecified number of symptoms, all causes other than the direct disorder may enter T_i^R forming at the most $(n-1)$ intermediate covers (if the set covering principle permits to form new intermediate covers). So, in any case, the number of intermediate covers added using the Realistic_ARM will be in the range of 1 and $(n-1)$. This is true for all the symptom-sets in a layer and all the layers in the knowledge base, which proves the theorem.

Remark 1: The number of covers generated by the Pure_ARM and the Realistic_ARM is equal when all the symptoms have only direct disorders (*i.e.*, in the case of *pathognomonic diseases*).

Theorem 2: Irredundant covers generated by Realistic_ARM remain the same for a given set of symptoms irrespective of the order in which the symptoms are explored.

Proof: In the process of reasoning, redundant covers may be generated and later removed or, following the set covering principles, not be generated at all. But, finally the irredundant covers remain the same irrespective of the order in which symptoms are explored. We demonstrate this by using an example.

Consider the scenario in layer i $(1 \leq i < N)$ with the following knowledge base depicted as in Fig. 1.

Suppose that $\{m_1, m_3, m_4\}$ are the given symptoms. Now, we show for the ordered sets $\{m_1, m_3, m_4\}$ and $\{m_4, m_1, m_3\}$ that the final irredundant covers remain the same, which is also true for other possible orderings.

For the symptom-set $\{m_1, m_3, m_4\}$, starting with symptom m_1, the intermediate cover-set T_i is $\{\{d_2\}, \{d_4\}\}$, which is already irredundant. When symptom m_3 is added, T_i is $\{\{d_2\}, \{d_4, d_2\}\}$, out of which, $\{d_2\}$ is the only irredundant cover. And, when m_4 is added, $\{d_2\}$ is already able to provide the explanation and remains as the final irredun-

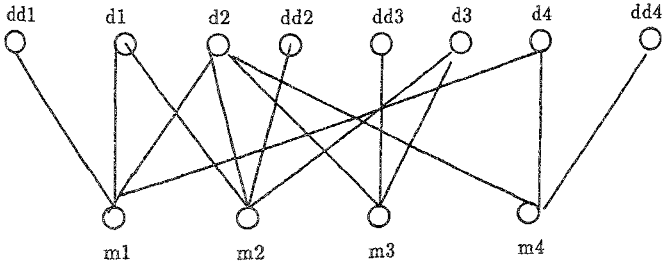

FIG. 1. Knowledge base in one of the layers of the hyper-bipartite network. The direct disorder of a manisfestation m_i is denoted as dd_i.

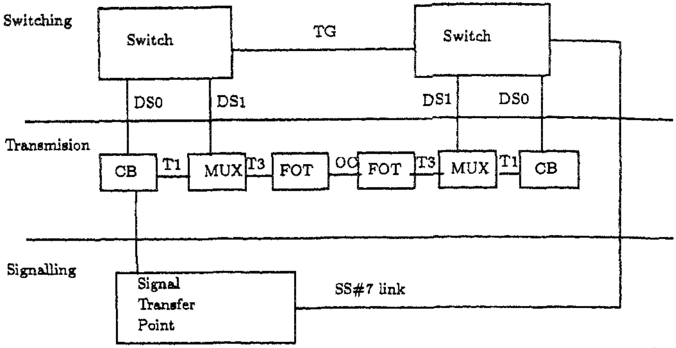

FIG. 2. Restricted communication network.

dant cover. (At this point, reinitialize T_i to $\{\emptyset\}$ since it has to hold intermediate covers for the next symptom-set). Now, consider the symptom-set $\{m_4; m_1, m_3\}$. Starting with symptom m_4, T_i is $\{\{d_2\}, \{d_4\}\}$ with all irredundant covers. When symptom m_1 is added, both the covers in T_i are able to provide the explanation. And when symptom m_3 is added, T_i is $\{\{d_2\}, \{d_4, d_2\}\}$, out of which $\{d_2\}$ is the only irredundant cover. This proves that the order of exploring the given symptoms in a layer does not affect the final irredundant covers.

Remark 2: As all the symptoms are available before generating the cover-set of that layer, we do not place the direct disorder of a symptom in one explanation since pre-specified number of symptoms are not available to support the symptom; and place the common disorder in obtaining the other explanation since the prespecified number of symptoms are available later.

4. Network fault management using Realistic_ARM

The fact that the Realistic_ARM is a compromise between the extreme cases of abductive and deductive reasoning models is utilized here to solve the communication network fault diagnosis problem. For a restricted communication network fault model, we demonstrate the use of realistic abductive reasoning model for fault diagnosis.

4.1. *The common channel signalling network fault management*

In the communication network under consideration[13–15] (see Fig. 2), two switches are connected *via* trunk groups, carrying T1 links. The T1 link is multiplexed through a multiplexer (MUX) to a T3 link which, in turn, is multiplexed through a fiber optic terminal (FOT) to an optical carrier (OC) signal. The switches are connected through CCITT signalling system #7 (SS #7) links to a signal-transfer point (STP). Signalling links are carried on DS0 channels, and are routed through a channel bank (CB) to the multiplexer and fiber optic terminal.

4.1.1. *Assumptions*

We consider a fault model with the following assumptions.

- management center receives alarms from all network elements; end-point switches, the transmission equipment, and the signal transfer point.
- the alarms could be because of cable cut or failure of one or more network element(s); needs precise diagnosis.
- there may be some missing information and the entire information may not be available at a time.

4.1.2. *The fault knowledge model*

The restricted communication network fault knowledge model[14,15] is constructed as a hyper-bipartite network (see Fig. 3). This maps the network fault knowledge on to a model suitable for the Realistic_ARM.

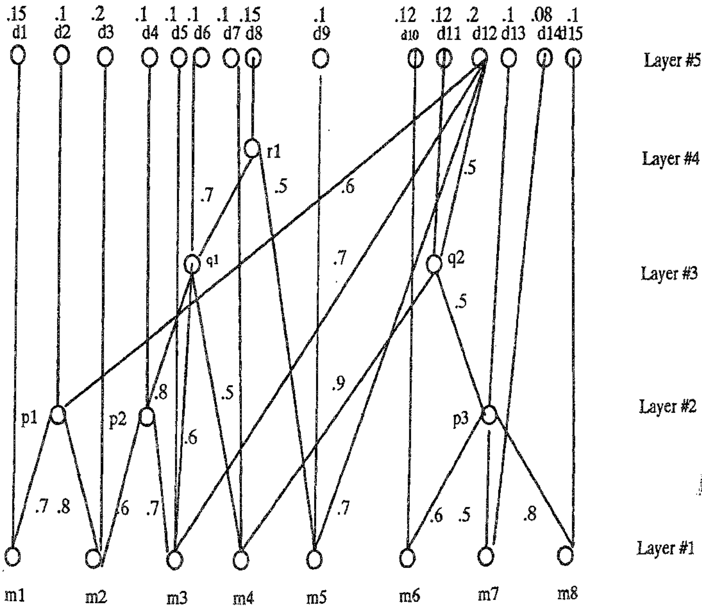

FIG. 3. Restricted communication network fault knowledge model. The plausibilities of the direct disorders are taken to be 1.0.

Legend

Layer #1

$m1$: Common memory alarm
$m2$: Trunk group alarm
$m3$: Facility interface unit (FIU) alarm
$m4$: MUX alarm
$m5$: FOT alarm
$m6$: SS#7 link alarm
$m7$: SS#7 interface (SSI) alarm
$m8$: Channel bank alarm

Layer #2

$p1$: Common memory failure
$p2$: FIU failure
$p3$: Channel bank signal failure

Layer #3

$q1$: MUX, carrying voice failure
$q2$: MUX, carrying signal failure

Layer #4

$r1$: FOT, carrying voice failure

Layer #5

$d1$-$d15$ except $d12$: direct disorders corresponding to symptoms
$d12$: FOT, carrying voice and signal failure

As proposed in the previous section, the communication network fault knowledge is classified into three categories, *viz.*, switching, transmission and signalling units. The network fault knowledge model constructed in the form of hyper-bipartite network will be transformed into a layered network and the inference mechanism proceeds from the bottom-most layer to the top-most layer.

For better understanding of the model, the following example illustrates symptom-set for the above knowledge base.

4.1.3. *An illustrative cover generation for CCS SS #7*

We describe the cover generation and the probability assignment of one sample set of symptoms which have two covers.

Consider the symptoms $\{m_3, m_4, m_5\}$ as a test case.

Case 1

At layer 1

In this layer, the symptom-set is $\{m_3, m_4, m_5\}$ and the cover-set is also $\{m_3, m_4, m_5\}$. Since the symptoms are the direct disorders of themselves, $M1 = M2 = M3 = M = 1.0$ and all the symptoms at layer 2 take the plausibility of 1.0.

At layer 2

Here the symptom-set is $\{m_3, m_4, m_5\}$ and the explanation is $\{q_1, m_5\}$.
$M1([q1, m5], [m3, m4, m5]) = [(1)\ (1-(1-0.6))]^*\ [(1)\ (1-(1-0.5))] = 0.3$
$M2([q1, m5], [p2]) = (1-0.8) = 0.2$
$M3 = 1.0$
$M = M1 * M2 * M3 = 0.3 * 0.2 * 1.0 = 0.06.$

At layer 3

The symptom-set is $\{q1, m5\}$ and the cover is $\{r1\}$.
$M1([r1], [q1, m5]) = [(0.06)^*(1(1-0.7))]^*\ [(1)^*\ (1-(1-0.5))] = 0.021$
$M2 = 1.0$
$M3 = 1.0$
$M = 0.021$

At layer 4

The symptom-set is $\{r1\}$ and the cover set is $\{d8\}$.
$M1 = 0.021 * (1-(1-1)) = 0.021$
$M2 = 1.0$
$$M3 = \frac{0.15}{1-0.15} = 0.17647$$
$M([d8], [m3, m4, m5], [p2]) = 0.0037$

Case 2

Similarly, for the symptom-set $\{m3, m4, m5\}$ in the other case, M until layer 3 is 1.0.

At layer 4

$M1([d7, d12], [m3, m4, m5]) = [(1)\ (1-(1-0.7))]^*\ [(1)\ (1-(1-0.7))] = 0.49$
$M2([d7, d12], [p1, q2]) = (1-0.6)^*\ (1-0.5) = 0.2$
$$M3 = \frac{0.1}{1-0.1} * \frac{0.2}{1-0.2} = 0.02777$$
$M([d7, d12], [m3, m4, m5], [p1, q2]) = 0.49 * 0.2 * 0.02777 = 0.0027$

Thus the cover $\{d8\}$ is more plausible than the cover $\{d7, d12\}$.

At this juncture, the fault manager would recommend to attend the more plausible fault to correct the hard failure.

Table I
Some results obtained by using the Realistic_ARM for communication network fault diagnostic problem

Sl. no.	*Observed symptoms*	*Covers with plausibilities*
1.	$\{m2, m8\}$	$\{d3, d15\}(0.027778)$
2.	$\{m3, m4, m5\}$	$\{d8\}, \{d7, d12\}$
3.	$\{m4, m6, m8\}$	$\{d11\}(0.0147)$
4.	$\{p1, q2\}$	$\{d12\}(0.00675)$

4.1.4. *Results and discussion*

The algorithm, Realistic_ARM, is run for various sets of symptoms and some of the results are given in Table I. The number of other symptoms required to conclude a disorder of a symptom is set to 1.

For a discussion on the results obtained in Table I, consider a case where the alarms are observed from *trunk group* and *channel bank*, ($\{m2, m8\}$). If no other relevant symptom is found to support this, it is appropriate to conclude that "only those two units are not working" rather than assuming one or a combination of "*FIU failed*"/"*failure of the multiplexer, carrying voice*"/"*FOT failure*"/"*failure of the multiplexer, carrying signal only*"/"*failure of the FOT, carrying voice and signal*". Similarly, when there are alarms from *multiplexer*, *SS7 link* and *channel bank* ($\{m4, m6, m8\}$), it is enough to conclude that "*failure of the multiplexer, carrying signal only*" without waiting for any more diagnostic information and proceed for isolating the causes of that fault. (In this restricted communication network fault knowledge model, we have not included such details.)

From Table I, it can be observed that the covers generated by the proposed model contain appropriate explanation for any given symptoms without much of extra guess. Otherwise, generating so many explanations is computationally expensive and, further, it requires elimination of inappropriate covers using some heuristic method. The proposed model avoids these problems and still makes appropriate guess.

5. The network performance management problem

The aim of network performance management is to tune the network parameters in real time so that the network can be restored to normal from the degraded state[16]. In the communication networks scenario, some information may be missing and all the information that is required for fault identification may not be available at the time of diagnosis. If the deductive reasoning mechanism is applied to such a problem, the fault cannot be identified since all the symptoms may not be present. At the same time, the abductive reasoning approach will result in a large number of unwanted explanations for a given set of symptoms. Subsequently it will be very difficult to pinpointedly identify the explanation that has caused the degradation in the network performance. The realistic abductive model can be found to satisfy the requirements of the problem[17].

The prespecified number of symptoms required to support a given symptom before concluding a fault is a variable. This can be set based on the incremental step in which the performance needs to be tuned. Intermediate layer of diagnostic knowledge base enables a hypothesis to be given in any form, namely, from the lower layers as a result of reasoning process or as a symptom in the respective layer. The direct disorder to every symptom, whether in the bottom-most layer or the intermediate, allows the fault to be concluded very precisely without waiting for the rest of the symptoms to conclude the faults in the top-most layer.

The realistic abductive reasoning model in its original form allows the reasoning mechanism to query back the user (here, the managed nodes) to confirm the missing symptoms before concluding any fault. But, since performance tuning cannot be deferred for such a long time before all the required symptoms are obtained, this can be relaxed since the model allows some tolerance on the number of symptoms required to conclude reason for degradation in the network performance.

By suitably constructing the network fault knowledge model required for performance tuning, this model can be found to give very good results for the problem. A case study of Ethernet performance management, discussed in the following, illustrates this approach.

5.1. *Case study: Ethernet performance management model*

In this section, we consider a restricted Ethernet model to illustrate the ideas presented in this work. We assume that the reader is aware of Ethernet operation[18,19].

We consider an Ethernet performance management model with the following assumptions.

- The information that needs to be *monitored* for the purpose of performance tuning is collected from the stations and the channel. And, that information, which is beyond the normal (both above and below the normal limits), is reported as symptoms.
- Some monitoring information like *load is normal* and *collisions are within the range* are included to support the diagnostic process by eliminating unnecessary fault sets which otherwise raise false alarms.
- there may be some missing information and the entire information may not be available at the time of diagnosis.

5.1.1. *The Ethernet performance management knowledge model*

The Ethernet performance management knowledge base[18-22] is constructed as a hyper-bipartite network (see Fig. 4). This maps the network performance management knowledge on to a model suitable for the Realistic_ARM.

Legend

Layer #1

1. Packet loss below normal	11. Large packets normal
2. Packet loss normal	12. Large packets above normal

3. Packet loss above normal
4. Load below normal
5. Load normal
6. Load above normal
7. Collisions below normal
8. Collisions normal
9. Collisions above normal
10. Large packets below normal
13. Small packets below normal
14. Small packets normal
15. Small packets above normal
16. Broadcast packets normal
17. Broadcast packets above normal
18. Packet loss on spine above normal
19. Load on spine normal
20. Load on spine above normal

Layer #2

1. Light traffic
2. Heavy traffic
5. Preambles are many
6. Broadcast packets are many

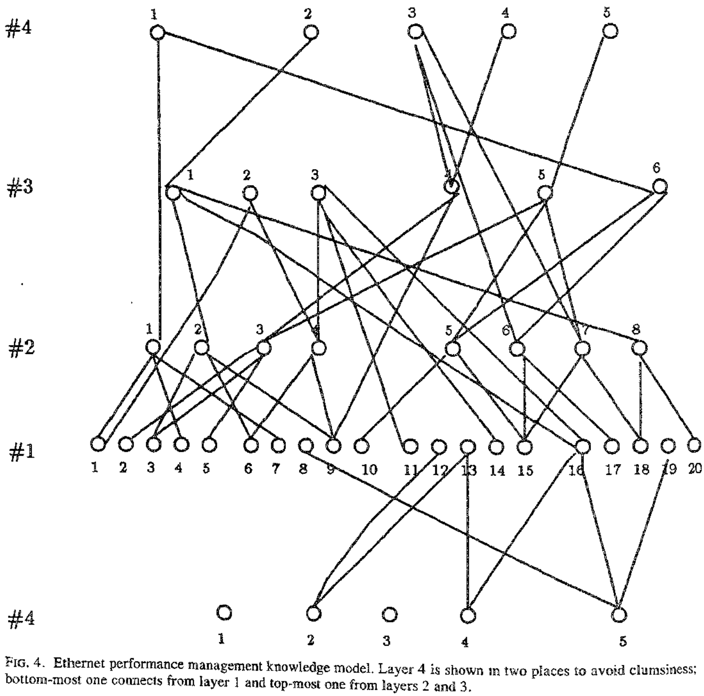

FIG. 4. Ethernet performance management knowledge model. Layer 4 is shown in two places to avoid clumsiness; bottom-most one connects from layer 1 and top-most one from layers 2 and 3.

3. Buffers are insufficient
4. Users are many
7. Spine with many small packets
8. Heavy traffic on spine

Layer #3

1. (F1) Babbling node; (Remedy, R1): Faulty Ethernet card, report to the network manager
2. (F2) Hardware problem; (Remedy, R2): Request the network manager to initiate fault diagnosis measures
3. (F3) Jabbering node; (Remedy, R3): Ensure many packets are not above the specified size
4. Too many retransmissions
5. Underutilization of channel as many small packets are in use
6. Attempt for too many broadcasts

Layer #4

1. (F4) Bridge down; (Remedy, R4): Report to the network manager
2. (F5) Network paging; (Remedy, R5): Allocate more primary memory to the required nodes.
3. (F6) Broadcast storm; (Remedy, R6): Selectively control the broadcast packets
4. (F7) Bad tap; (Remedy, R7): Report to the network manager along with the specified tap
5. (F8) Runt storm; (Remedy, R8): Ensure many packets are not below the specified size

The fault knowledge base, constructed in the form of a hyper-bipartite network will be transformed into a layered network for a given diagnostic problem. The inference mechanism proceeds from the bottom-most to the top-most layer to find a solution for a given set of symptoms.

5.1.2. *Results*

The algorithm, Realistic_ARM, was run for various sets of symptoms (from layer 1 of Fig. 4) and some results are given in Table II. The prespecified number of symptoms required to support any symptom before concluding a fault is set to 1.

Table II
Sample results for Ethernet performance model

Sl. no.	*Symptoms*	*Suggested remedy*
1.	3, 6, 12, 18, 20	$\{R5\}$
2.	1, 4, 10, 15, 17	$\{R4\}$
3.	3, 9, 18, 20	$\{R1\}$
4.	10, 15, 16, 18	$\{R8\}$

6. Conclusion

Abductive reasoning is shown to be well suited for the specialized problems of network fault diagnosis. The diagnostic problem is then solved by using the realistic abductive reasoning model. The explanation provided by the model is appropriate and shall not have much of extra guess. When more than one explanation exists, plausibilities are assigned to each of them to explore in the decreasing order of plausibility till the real fault is isolated. The network performance degradation is considered as a special case of soft failures and is also solved using realistic abductive reasoning model. Two case studies of common channel signalling network fault management and the Ethernet performance management are discussed. The results obtained by the proposed model are quite encouraging.

References

1. Cassel, L. N., Patridge, C. and Westcott, J. Network management architectures and protocols: Problems and approaches, *IEEE J.*, 1989, SAC-7, 1104–1114.
2. Bernstein, L. and Yuhas, C. M. Expert systems in network management—the second revolution, *IEEE J.*, 1988, **SAC-6**, 784–787.
3. Cronk, R. N., Callahan, P. H. and Bernstein, L. Rule-based expert systems for network management and operations: an introduction, *IEEE Network*, 1988, 2, 7–21.
4. Marques, T. E. A symptom-driven expert system for isolating and correcting network faults, *IEEE Commun. Mag*, 1988, **COM-M-26**, 6–13.
5. Arun, R. and Venkataram, P. Knowledge-based trouble shooting in communication networks, *Proc. Symp. on Intelligent Systems*, Bangalore, India, November 1992, pp.20–24.
6. Pople, H. On the mechanization of abductive logic, *Proc. Int. J. Conf. on Artificial Intelligence*, 1973, pp.147–152.
7. Reggia, J. A., Nau, D., Wang, P. and Peng, Y. A formal model of diagnostic inference, *Inf. Sci.*, 1985, **37**, 227–285.
8. Peng, Y. and Reggia, J. A. Diagnostic problem-solving with causal chaining, *Int. J. Intell. Systems*, 1987, **2**, 395–406.
9. Goodall, S., Reggia, J. A., Tuhrim, S. and Chu, B. ABLOG: Improvements on an abductive approach to diagnostic problem solving, *Proc. Third Annual Expert Systems in Government Conf.*, Washington, DC, October 1987, pp.150–154.
10. Peng, Y. and Reggia, J. A. *Abductive inference models for diagnostic problem-solving*, 1990, Springer-Verlag.
11. Bylander, T., Allenmang, D., Tanner, M. C. and Josephson, J. R. The computational complexity of abduction, *Artif. Intell.*, 1991, **49**, 25–60.
12. Kumar, G. P. and Venkataram, P. A realistic model for diagnostic problem-solving using abductive reasoning based on parsimonious covering principle, *Proc. Third Turkish Conf. on Artificial Intelligence and Neural Networks*, Ankara, Turkey, June 1994, pp.1–10.
13. Freeman, R. L. *Telecommunication system engineering*, 1989, Wiley.

14. SCHLANGER, G. G. An overview of signalling system No. 7, *IEEE J.*, 1986, **SAC-4**, 360–365.

15. KHERADPIR, S., STINSON, W., VUCETIC, J. AND GERSHT, A. Real-time management of telephone operating company networks: Issues and approaches, *IEEE J.*, 1993, **SAC-11**, 1385–1403.

16. HAYES, S Analyzing network performance management, *IEEE Commun. Mag.*, 1993, **31**, 52–59.

17. KUMAR, G. P. AND VENKATARAM, P. Network performance management using realistic abductive reasoning, *IEEE/IFIP Int. Symp. on Integrated Network Management*, Santa Barbara, California, May 1995, pp.187–198.

18. METCAFE, R. M. AND BOGGS, D. R. Ethernet: Distributed packet switching for local computer networks, *Commun. ACM*, 1976, **19**, 395–404.

19. BOGGS, D. R., MOGUL, J. C AND KENT, C. A. Measured capacity of an Ethernet: Myths and reality, *Comput. Commun. Rev.*, 1988, 222–234.

20 FEATHER, F. E., SLEWLOREK, D AND MAXION, R. Fault detection in an Ethernet network using anomaly signature matching, *Comput. Commun. Rev.*, 1993, 279–288.

21. FEATHER, F. E. *Fault detection in an Ethernet network via anomaly detectors*, Ph. D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1992.

22. HARSEN, J. P. The use of multi-dimensional parametric behavior of a CSMA/CD network for network diagnosis, Ph. D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1992

Appendix I

Algorithm Realistic_ARM

Nomenclature

1. *temp_man* is a set of symptoms at the layer of inference. (By both, one of the covers of the previous layer and the symptoms of that layer.)
2. *prim_man* is a set of symptoms available at all the layers, holds the symptoms provided by the user minus the symptoms explored in all the previous layers (retained if the manifestation is present in the next layer because of dummy nodes created by *Build_Layered_Net*).
3. *sec_man* is a set of symptoms available at all the layers, holds all the symptoms that are provided by the user.
4. *More_Manifs*, a boolean, is TRUE if there are any more symptoms found to exist at a layer by either input or when asked interactively through common disorders of the existing symptoms. Otherwise it is FALSE.

Algorithm Realistic_ARM
{
  var *i*, *j*, pre_lay_cov_count: *int*;

```
Call procedure Build_Layered_Net;
Read the given symptoms into prim_man and sec_man.
C0 = {∅};
loop:
for(i = 1; i < N; i++)
{
    pre_lay_cov_count = |C(i-1)|; j = 0;
    For all the symptoms of layer i, query the related
    manifestations through common disorder and place them in prim_man.
    do
    {
        temp_man = ∅;
        if(|C(i-1)| > 0)
        Get jth cover of layer (i - 1) into temp_man.
    Append symptoms of layer i that are present in prim_man to temp_man.
 Ti = Gen_Covers(temp_man);/*Generate covers for the symptom(s) present in
        temp_man.*/
    Call the procedure Update_Prob(Ti, temp_man);
    Ci = append(Ci, Ti);
 }while(--pre_lay_cov_count > 0);
 Delete symptoms of layer i from prim_man if they do not
        exist in layer (i + 1).
 Ci = Gen_Irr_Covers(Ci)//Generate the irredundant covers for layer i.
}//end of for(i < N, no. of layers)
Append the disorders of layer N present in layer prim_man to each of the covers if they
    do not already exist.
CN = Gen_Irr_Covers(CN);
Call the procedure Update_Prob(CN, prim_man);
Delete the symptoms of layer N from prim_man.
if(some symptoms are still left in prim_man)
{
    prim_man = ∅
    Copy sec_man to prim_man and goto "loop".
}
Output the final covers, CN.
Suggest suitable remedies for CN in the decreasing order of plausibility.
}//end of algorithm Realistic_ARM

Procedure Build_Layered_Net
{
    Retain the nodes of the hyper-bipartite network.
    For each layer i, (1 ≤ i ≤ (N - 2)), of hyper-bipartite network:
```

```
if there is a link from layer i to layer (i + 1), retain the same in the layered net-
    work.
if there is a link (say l_{h_m,h_n}) from manifestation/hypothesis at layer i to hypothe-
    sis/disorder at layer (i + k), k > 1, replace it by creating a dummy node with
    the name same as h_m at all the intermediate layers and connect them.
}//end of procedure Build_Layered_Net

Function Gen_Covers(temp_man)
{
  var k, p, q, u, v: int;
   cov_added: boolean;
  T_i = {Ø};
  for(k = 0; k < |temp-man|; k++)
  {
   if(k == 0)
   {
     for(u = 0; u < v, no. of disorders of kth symptom; u++)
     {
     if(uth disorder of symptom k is supported by a prespecified number of symp-
         toms)
      t_{|T_i|++}^i = {uth disorder};
    }
    if(|T_i| == 0)
      t_{|T_i|++} = {direct disorder of symptom k};
   }//end of if(k == 0)
   else//if(k ≠ 0)
   {
    q = |T_i|;
    for(p = 0; p < q; p++)
    {
      cov_added = FALSE.
      for(u = 0; u < v, no. of disorders of symptom k; u++)
    {
      if(uth disorder of symptom k is supported by a prespecified number of symp-
          toms and ∈ t_p^i) /* t_p^i is already a cover for symptom k */
      {
       goto next_cover;
      }
```

```
        }//end of for(u < v)
        for(u = 0; u < v, no. of disorders of symptom k; u++)
        {
          if(uth disorder of symptom k is supported by a prespecified number of symp-
              toms)
          {
            t_{|T_i|++}^i = append(t_p^i, uth disorder);
            cov_added = TRUE;
          }
        }//end of for(u < v)
        if(cov_added == TRUE)
        {
          Mark t_p^i for deletion.
          goto next_cover;
        }
        t_{|T_i|}^i = append(t_p^i, direct disorder of symptom k);
        next_cover: ;
       }//end of for(p < q)
       Delete those covers marked for deletion from T_i and update |T_i|.
     }//end of else if(k ≠ 0)
     T_i = Gen_Irr_Covers(T_i); //Make irredundant after each symptom is explored
   }//end of for(k < |temp_man|)
   return T_i;
}//end of function Gen_Covers

Function Gen_Irr_Covers(T_i)
{
   var u : int;
   for(u = 0; u < |T_i|; u++)
   {
      if(t_u^i is unmarked and is a superset of any other cover in T_i)
        Mark t_u^i for deletion;
   }
   Remove the covers that are marked for deletion from T_i.
   return(T_i)
}//end of function Gen_Irr_Covers

Function Update_Prob(T_i, temp_man)
{
```

var j, k, l, n: *int*;
$M1, M2, M3, M$: *float*;
Cover, M^+, M^-: *type cover*;
for($n = 0$; $n < |T_i|$; n++)
{
Cover = nth element of T_i.
M^+ = temp_man $\bigcap$ effects(all elements of Cover);
M^- = effects(all elements of Cover) - symptoms observed to be present.

$$M1 = \prod_{[m_j \in M^+]} P(m_j) \cdot \left(1 - \prod_{[d_k \in E]} (1 - p_{kj})\right).$$

$$M2 = \prod_{[d_k \in E]} \prod_{[m_l \in M^-]} (1 - p_{kl}).$$

$$M3 = \prod_{[d_k \in E]} \frac{P(d_k)}{1 - P(d_k)}.$$

$M = M1 * M2 * M3$.
For each *element* of *Cover*, the measure of P(element) as a symptom for next layer is max(measures obtained to its credit so far).
}//end of for($n < |T_i|$)
}//end of function Update_Prob

Appendix II

Algorithm Pure_ARM

{
var i : *int*;
Call procedure Build_Layered_Net_Old; //refer to Peng and Reggia[10]
// accommodates dummy nodes to accept symptoms into the covers at any time
$C_1 = \{\emptyset\}$;
for($i = 1$; $i < N$; i++)
{
Query the related manifestations through common disorders of its existing symptoms of layer i.
while(More_Manifs)//loops for each symptom
{
C_{i+1} = Gen_Covers_Old(C_i); //using unrestricted abductive inference
C_{i+1} = Gen_Irr_Covers(C_{i+1});
}// end of while(More_Manifs)
}//end of for($i < N$)
Output C_N;
}//end of algorithm Pure_ARM