FAULT DIAGNOSIS FOR LOCAL AREA NETWORK ENVIRONMENTS

Volnys Borges Bernal
volnys@lsi.usp.br
LSI-EPUSP
University of São Paulo
Av. Prof. Luciano Gualberto,
trav.3, 158. São Paulo, SP, Brazil
05508-900

Leliane N. de Barros
leliane@ime.usp.br
IME - BCC
University of São Paulo
Rua do Matão, 1010
São Paulo, SP, Brazil
04717-020

Marilza Lemos
mlemos@lsi.usp.br
LSI - EPUSP
University of São Paulo
Av. Prof. Luciano Gualberto,
trav.3, 158. São Paulo, SP, Brazil
05508-900

Jacques Wainer
wainer@dcc.unicamp.br
Instituto de Computação
University of Campinas
Campinas, SP, Brazil

Abstract

Particular attention has been given to propose systems to solve network management tasks, especially for the fault diagnosis and performance management. The construction of such systems requires an intense process of modeling the network environment through interviews with experts in the network management area as well as the use of Artificial Intelligence techniques. The aim of this paper is to specify Problem Solving Methods for the diagnosis of communication network faults. We claim that the AI approach called Model-based diagnosis provides a foundation for exchanging behavioral, structural and control information between the subtasks of such complex systems. We also show what are the main aspects to be considered when constructing such systems, namely: how to build network models (manually or automatically); how to model an appropriate problem solving method to each class of network faults; how to identify the type of interaction between the diagnosis system and a network status gathering system, such as a management platform; how to construct the communication interfaces among several systems, etc. Finally, this work presents two prototypes of diagnosis systems: one for configuration faults and another for communication faults of conventional TCP/IP local area networks. [*]

1. Introduction

The growing complexity of the network management task and the lack of specialized professionals to perform it have shown the need to develop supporting tools for the network administrator task. This necessity can be verified with the growing number of network management platforms in the market and self-manageable network applications. In the research field, Diagnostic Tools for Network Fault Management are often proposed. The construction of these systems requires an intense process of knowledge acquisition from the experts in the area as well as the use of Artificial Intelligence (AI) techniques. Several approaches can be found in the literature to solve problems on fault diagnosis in communication networks such as: rule-based diagnosis, case-based diagnosis, neural-nets-based diagnosis and model-based diagnosis. The most common approach is the rule-based technique [10] [15] [2] [17] which has been very useful to solve routine problems. However, the knowledge embedded in such systems is specific to a particular configuration of network and only applied to networks whose configuration is rarely modified or updated. A rule-based system is difficult to be maintained due to changes in the network. Some works present solving methods for diagnosis that reason with probabilistic information about network faults. The difficulties in using this approach are the attribution of the fault probabilities to the managed objects and the identification of the dependencies among the objects [13] [4] [7]. We can find some works applying Neural Network AI technique to solve the diagnostic fault problem, for example, by doing the alarm correlation in a cellular phone network [19]. The weakness in this approach is the need of training the Neural Network every time that the managed Communication Network changes. Fault diagnosis systems using the Case-based AI approach can be found in the literature [8] [14] where cases can be seen as trouble tickets. This approach can be interesting only if trouble tickets are generated from a diagnostic system

[*] This material is based on work supported by the ABC Bull S.A. Telematic.

capable of supplying good explanations for every diagnostic in the case base. Recently, we have found some work applying the Model Based Diagnostic approach (MBD) to the network management domain. This approach proposes the construction of Knowledge Based Systems (KBS) through the specification of two basic structures: Domain Models and Problem Solving Methods (PSM) [1] [5] [6] [16] [18]. A uniform representation for network models allows matching of components and variants among different views of the network domain. We claim that Model Based Diagnostic provides a foundation for exchanging behavioral, structural and control information between the subtasks of complex systems.

In this work, we apply the model based approach to perform the diagnosis communication fault in the network domain. Therefore, the proposed method can consult models that represent the network in its multiple aspects, such as: configuration model (equipments, connections, repetition domains, subnetworks, routing tables, etc.), faulty state causal model and so on. The most important characteristic of using the Model Based approach for the network management domain is that the proposed method will be independent of the managed network, guaranteeing reusability of the system and maintenance: if the communication network changes, only the corresponding models should be modified [13]. Another important feature of the Model Based approach is to provide systems that can explain its conclusions.

One of the most common faults that can occur in a network is the lack of communication among network elements, usually caused by problems in the connections or in the elements themselves. A network diagnosis system should be able to solve communication fault before trying to solve other kinds of faults since, in the in-band model of the SNMP management, it should always be possible the communication between the manager and the SNMP agents.

A well known problem related to the implementation of a diagnostic system for Network Management is the complexity of the network environment since a regular network can have a variety of types of hardware components and a large number of them. The construction of network models involves the identification of all the necessary knowledge for the diagnostic task. Besides the identification of the necessary models, they have to be organized in a certain way that makes the existents relationship clear. In this work, it is proposed a network knowledge organization in levels, named ***configuration model*** (Section 4), that shows these relationships explicitly.

The acquisition and construction of network models can be done through an automatic discovery system (Section 7). It has been developed a Configuration Diagnosis System which can detect configuration errors during the acquisition and construction of the network models (Section 7).

In the following sections, it is presented all the aspects involved in the construction of these systems, as follows: in Section 2, it is seen a Diagnosis System Classification according to the type of interaction with the network environment; Section 3 introduces the Model Based AI approach; Section 4 describes the network models used by the diagnostic system; Section 5 specifies the Diagnostic Problem Solving Method developed in this work; Section 6 presents some diagnostic algorithm that corresponds to the diagnostic classes pointed out in Section 2; Section 7 describes the architecture of the global system and its subsystems: the Communication Fault Diagnosis System (CFDS), the Network Discovery System (NDS), the Configuration Model Constructor System (CMCS), the Configuration Diagnosis System and the Network Status Gathering System (NSGS). In Section 8, our contributions and the future of this work are discussed.

2. Diagnosis system classification according to their interaction with environment

According to what we have mentioned in section 2, a model-based diagnosis system can use one or more network models in order to accomplish the fault diagnosis task. Besides that, the diagnosis system needs to get information about the current environment status. It consists of a set of observations, some of them formed by symptoms (abnormal observations).

There are two important aspects involved in the process of information gathering (observation): the kind of data exchanged and the possible ways that the diagnosis system interacts with the network environment to collect this information.

There are three possibilities of exchanging data that can be obtained from the management platform by the diagnosis system: raw value, state value and transition value. **Raw values** are like those usually found on SNMP objects that do not indicate a normal or an abnormal state. Normally, those values have no significance alone. A management platform can also supply to the diagnosis system the **faulty state** associated to the object. For that, it is usually necessary to define expressions over values of one or more SNMP objects and the thresholds A threshold may indicate if it is in a normal or in a faulty state. **Transition values** correspond to an alarm generated by the management platform when it detects the occurrence of a state transition of a managed object. Examples of alarms generated in state transitions are those associated to the state of the manager's communication (management platform) with the network equipments. In this paper, it is proposed a method that reasons about state transition based alarms.

Based on this discussion it was identified three kind of Diagnostic Systems, described as follows:

2.1 Active Diagnostic System

An Active Diagnostic System obtains network information in a synchronous way, that is, it requests data from the network environment. This kind of system has a high degree of autonomy and is able to decide and request additional information from network environment during the diagnosis process. This characteristic can be very convenient due to the great number of alarms that can eventually exist in the network. The system may detect an initial symptom to start its task and then, it consults the network models and requests, in an autonomous way, the necessary information about the network to reason about the diagnostic.

2.2 State Based Passive Diagnostic System

A State Based Passive Diagnostic System waits for the information to come from the management platform, that means, it does not make any information request. Typically, a network platform gathers information and turns it available (i.e., events, alarms, etc.) through a polling mechanism. A polling cycle is a periodic sample usually performed by the network management platform which sweeps the network devices. Each type of observation may have a different polling cycle. For each type of observation, the platform sweeps periodically part of or all the network elements, performing tests and requests. A State Based Passive Diagnostic System uses only this information, provided by the platform, to reason about a diagnosis, that is, the network observation is exclusively controlled by the platform with no interference from the diagnostic system.

2.3 Transition Based Passive Diagnostic System

Some management platforms accomplish probes on the network, but they do not turn available the result of each probe. Instead of this, they just turn available the transitions among states. For example, if a network machine is not working, only one faulty state alarm is sent, independently of the number of polling cycles accomplished by the management platform. A related normal state alarm is only generated when the management platform detects that the machine is working again.

The behavior of a Transition Based Passive Diagnostic System is a variation of the State Based Passive Diagnostic System described above in the sense that it also uses just the information provided by the management platform. The difference is that the available information is the state transitions.

3. Model based Diagnostic Systems

The basic paradigm of the model based diagnosis can be understood as the interaction of observations and predictions as in Figure 1 [Davis & Hamscher, 88]. In this figure, there is the actual device in one side, which in this work corresponds to the network environment, and the device model in the other side, i.e., the network model. This model typically describes the components of the system, their connections and behavior. The fault behavior of components and functions of the device also belong to the device model. A difference between an observation and a prediction is called a discrepancy. A match between an observation and a prediction is called corroboration. Both discrepancies and corroborations are used to identify which parts of the network are incorrect.

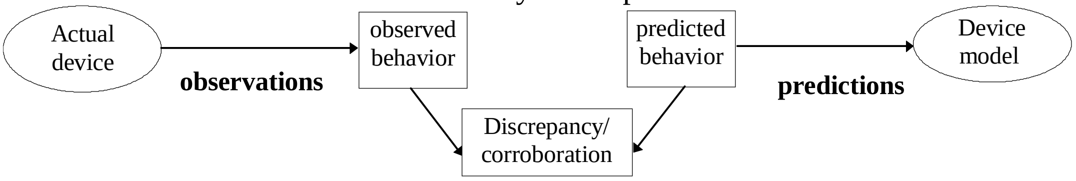

Figure 1. Model based diagnosis viewed as the interaction between observations and predictions.

Based on the above model-based diagnostic strategy, one can specify a variety of diagnostic problem solving methods. An important contribution of the Knowledge Engineering AI field is the idea of Problem Solving Methods (PSM) [6] to solve a specific class of problems (i.e. diagnostic, configuration, planning, etc.).

4. Identifying and building network models

One of the problems related to the implementation of a diagnostic system for Network Management is the complexity of the network environment. This is because a regular network can have: a variety of types of hardware components and a large number of them; there are different types of software components (protocols, operating systems, services and applications); the equipments and connections may be changed; and some network protocols are based on dynamic configuration.

4.1 The configuration Model

In spite of the complexity, in order to reason about a fault in a communication network, the necessary knowledge about the network environment should be available to the system. Besides the availability, the knowledge must be well organized in such a way that one can clearly see the relationship between components. In this work, it is proposed a network knowledge organization in levels that makes these relationships explicit. The ***Configuration Model*** comprises a set of configuration levels described below:

Spatial distribution level: it describes how the equipments are spread over the buildings and rooms. In this case, the relationship is the neighborhood and location of equipments. This kind of information is important to diagnose building power-off or noise problems.

Equipment level: it represents equipments, their ports and the connections among them. It also relates the ports, interface names and MAC addresses. There are some relationships like the set of ports of each equipment, the MAC address assined to each port, the connections among the equipments, and so on. This description can help to diagnose communication faults associated to ports or connection problems.

Network element level: in this level, it is possible to represent the role of each equipment in the system. There are some equipments that have more than one role, for instance, a router with some ports configured as a bridge or a computer with a hub board. In this case, it is necessary to decompose the equipments in other entities, called *Network Elements*, each one with a clear role.

Repetition domain level: it allows to represent the repetition domains of the network and how they are interconnected. A repetition domain is a set of equipment ports to which an Ethernet packet is unconditionally repeated. Thus, the main relationship is which equipment port belongs to a repetition domain. This level is particularly important to diagnose which equipment interfaces are affected when there is an overload on a repetition domain.

Broadcast domain level: it represents the broadcast domains of the network and how they are interconnected. A broadcast domain is a set of equipment ports to which an Ethernet broadcast packet is unconditionally repeated. Thus, the main relationship is which equipment ports belong to a broadcast domain. This level is important to identify which equipment interfaces may cause or are affected by high level broadcast packets in a broadcast domain.

Subnetwork domain level: it represents the subnetworks and how they are interconnected. It also has information about the routing table of each equipment and IP-MAC addresses relationships. A subnetwork domain level is a set of IP interfaces that belongs to the same subnetwork.

Name domain level*:* it describes how the names are assigned to IP addresses and how they are grouped into an Internet Domain Name. A name domain is a set of host names that belongs to the same Internet Domain Name.

Services and application level*:* This is a generic level and may be decomposed into several configuration modules. It is intended to model necessary services and applications.

It is possible to view a generic TCP/IP local area network in these levels. In fact, the well known layers of the TCP/IP stack have driven us to organize the configuration model in the above levels. Table 1 shows how the network representation levels can be mapped to the TCP/IP network layers.

Configuration Model Level	TCP/IP layers
Name Domain Level	Application
Subnetwork Domain Level	Internet
Broadcast Domain Level	Interface
Repetition Domain Level	Interface
Network Element Level	Interface, Internet
Equipment Level	Interface
Spatial Level	---

Table 1. A mapping between the TCP/IP layers and the network models proposed in this work.

The configuration levels identified in this work describe the necessary knowledge for a range of diagnostic problems. In particular, it was used for communication fault diagnosis caused by equipment or cable problem (bandwidth and latency problems can also cause communication faults). In this case, it is only necessary to use knowledge about the Equipment Level, Network Element Level and Subnetwork Domain Level. In this work, the Configuration Model will only contain the levels that are important to solve this kind of communication problems. It is also important to notice that although the proposed Configuration Model allows to model a generic TCP/IP network environment, this work initially focuses the diagnostic of communication faultin a standard TCP/IP local area network.

4.1.1 The Network Element Level

As an equipment can have more than one role, it is introduced here the "Network Element" level. If some equipment has more than one role, in this level it is represented as several network elements, one for each role. We are considering the following network element roles: router, bridge, repeater and computer. Each one of these network element roles has one basic function.

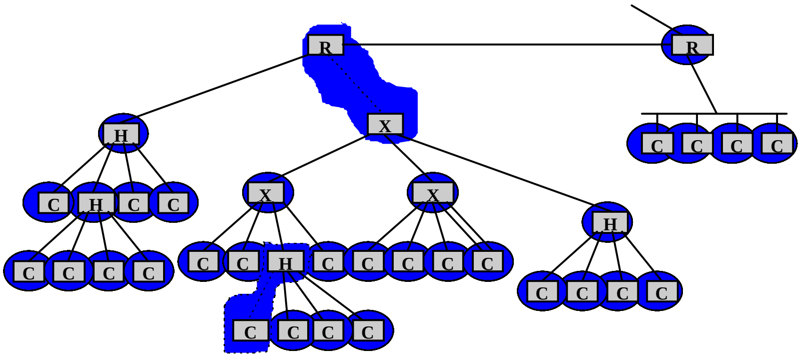

Figure 2. An example of a graphical representation of the topology obtained from the Network Element Level.

There are some relationship hold by this level: equipment ↔ network elements, port ↔ MAC and connections between network elements. The last one is obtained from the equipment level connections, adding the logical or internal connections among network elements of the same equipment.

4.1.2 The subnetwork domain level

This level represents the configuration of each subnetwork. For each network it is maintained the network address, the netmask, the broadcast address and the existent IP address. Each subnetwork may be mapped into a broadcast domain. Each broadcast domain may be shared by one or more configured subnetworks, as shown in figure 3.

The following information is part of this level: subnetwork configuration, subnetwork ↔ IPs relationships, IP ↔ MAC relationship and equipment routing tables.

This information allows us to represent a network environment over the subnetwork level view.

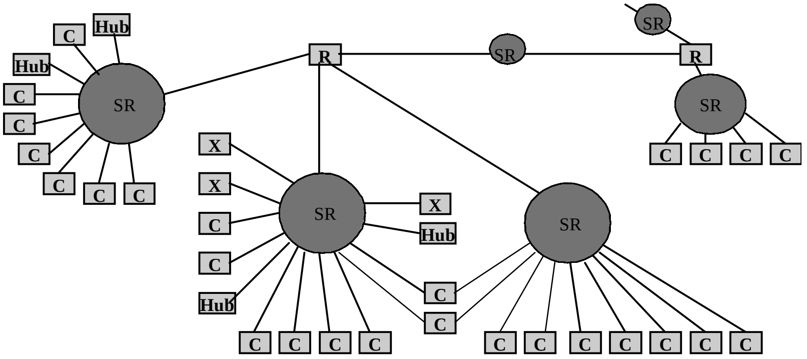

Figure 3. An example of a graphical repreentation of the topology obtained from the Subnetwork Domain Level.

4.2 Communication Path Model

The Communication Path Model corresponds to the human manager's common sense knowledge about communication faults or performance problems in the network. Particularly, this model specifies how to analyze paths of communication between the manager and the network elements in the presence of a faulty element. Reasoning about faults involves an important matter: the assumption of single or multiple faults in the domain. Therefore, the Path Intersection Model depends on which assumption is considered in the network in terms of single or multiple faults.

Even though the multiple fault assumption seems to be the real situation, the single fault assumption can deal with a number of interesting cases where independent sets of symptoms can be identified for multiple simultaneous faults. A diagnostic system, which makes the multiple fault assumption, can be more powerful but rather more complex to be built. Therefore, in this paper, we will assume that only a single fault occurs in one complete polling cycle. As it is defined above, in a Transition Based Passive Diagnostic System, which makes the assumption of a single fault, it is only necessary to analyze all the alarms of *unreachable* type in one single polling cycle, starting from the first alarm that has been detected until the end of the cycle, since all the consequences of a possible fault will be detected within one cycle (all the network elements will be checked by the manager). Multiple causes might be found by the diagnostic system which could deal with independent faults.

When a diagnostic system receives an "unreachable" alarm that indicates that a particular network element E is not responding, it can conclude that either E or some element in the path between the network manager and E is faulty. This set of network elements is called a hypothesis, that can be explained as a set of network elements in which at least one of them is faulty. Further alarms about other devices must be correlated with the current hypothesis. Below, six of the nine correlation rules used in this work are described:

Rule 1: An abnormal observation confirms the hypothesis and reduces it. The path between the manager and an observed device is faulty and there is an intersection with the current hypothesis. In this case, the new hypothesis is the intersection path.

Rule 2: An abnormal observation confirms the hypothesis but does not reduce it. The path between the manager and an observed device is faulty and the intersection with the current hypothesis is the hypothesis itself. In this case, the hypothesis remains the same.

Rule 3: An abnormal observation does not confirm the hypothesis and shows a new possible fault. The path between the manager and an observed device is faulty and there is no intersection with the hypothesis. Under the assumption of a single fault, this case corresponds to independent faults which can have an independent diagnostic, e.g., each hypothesis will only contain one fault. This is equivalent to assume a minimum number of faults: if two faults can explain all symptoms, we will assume that two faults exist.

Rule 4: A normal observation is independent and does not reduce the hypothesis. The path between the manager and an observed device is all right and there is no intersection with the current hypothesis. In this case, the hypothesis remains the same.

Rule 5: A normal observation contradicts part of the hypothesis. The path between the manager and an observed device is all right and there is an intersection with the current hypothesis. In this case, the observation eliminates part of the hypothesis, e.g., the intersection path.

Rule 6: A normal observation contradicts all of the hypothesis. The path between the manager and an observed device is all right and the hypothesis is the intersection itself. In this case, the observation deletes the hypothesis and the original fault can be considered as intermittent.

The knowledge described above defines part of a theory about how to process the result of a communication path which will be used by the Problem Solving Method specified in the next Section.

5. The Problem Solving Method

According to the Knowledge Engineering AI field, a general diagnostic Problem Solving Method [5] [6] decomposes the diagnosis task into three subtasks [3], as follows (see figure 4):

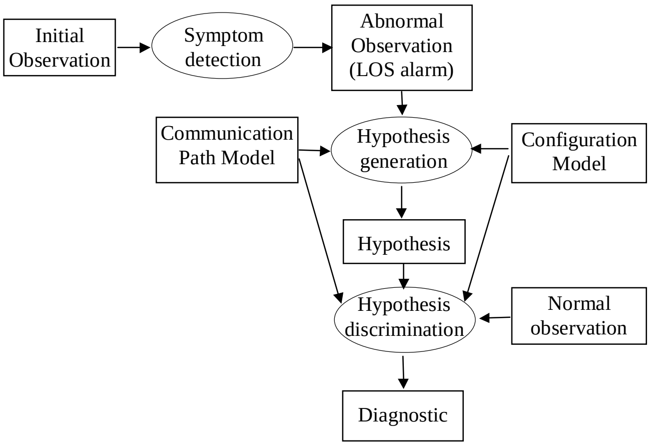

Figure 4. Data flow of the diagnostic method.

Symptom Detection: it detects symptoms starting from observations done on the device. A symptom corresponds to some abnormal observation. In the Communication Fault Diagnostic System (CFDS), the symptoms correspond to all the alarms of unreachable type detected by the manager (abnormal observation).

Hypothesis Generation: when there is a set of symptoms, this subtask determines possible causes (diagnostic) that explain their presence. Using the Configuration Model, the CFDS calculates the set of network paths used by the manager to communicate with each NE corresponding to the unreachable alarms in the symptoms set. These paths correspond to the initial hypotheses set. For every two communication paths (hypotheses), this task uses the Communication Path Model to compare the observations with the model and generate new hypotheses.

Hypothesis Discrimination: it analyzes the set of hypotheses generated by the previous task in order to determine the most probable one that is the best explanation for the observations. For that, the task can request additional observation on the device in order to get more information. In the CFDS, additional observation corresponds to all the devices that have no alarm of type unreachable associated to them, that is called *normal observation*. For this normal observation set, the CFDS calculates the new network paths using the Configuration Model. Finally, the Network Path Model is used to compare the observations with the model in order to discriminate the hypotheses.

5.1 The calculus of the communication path

The two diagnostic subtasks *Symptom Detection* and *Hypothesis Generation* have to execute the calculus of the communication path. This subtask starts from consulting the route table and identifying the routing elements sequence between the Network Management Platform and the target network device in both directions. The “routing elements sequence” are formed by these routing elements, including the Network Management Platform and the target network. The system then identifies the network elements between the “routing elements sequence” that forms the communication path.

6. Diagnostic Methods

The diagnostic methods depend on what kind of information the diagnostic system receives from the subsystem of Network Status Gathering System. In the next section, we will specify three diagnostic algorithms: the Active, the State Based Passive and the Transition Based Passive Method.

6.1 Active Diagnostic Method

In this method, the diagnostic system is started by an unreachable alarm from the Network Status Gathering System. After that, the diagnostic system controls the queries to the network environment, that is, the diagnostic system can make observations according to its reasoning needs.

```
The subsystem is started with the information that S is unreachable
    h = path(S)
    for each Elem $\in$ h (chosen intelligently)
        p = path(Elem)
        query Elem
        if Elem is reachable
            $h = h - (h \cap p)$
        else if Elem is unreachable
            $h = h \cap p$
    print diagnostic that “at least one element of h is faulty”
```

6.2 State Based Passive Diagnostic Method

This method is used when the Network Status Gathering System sends the result of each polling to the diagnostic system. Thus, for each network element at the appropriate moment in the polling cycle, the diagnostic system receives the information that either the element is reachable (responds to the query) or unreachable (does not respond to the query). The algorithm uses the following data:

- h is a hypothesis;
- H is a set of current hypotheses;
- Path (S) is the set of network elements and connections in the path from the network management system to the element S and back.

```
$H = \phi$
For each alarm related to the element S
    C = Path(S)
    If S is unreachable then
        If there is $h \in H \mid h \cap C \neq \phi$ then
            $h = h \cap C$
        else
            $H = H \cup \{ C \}$
    else if S is reachable
        for all such that $h \in H$ and $h \cap C \neq \phi$
            $h = h - h \cap C$
At the appropriate moments, print the diagnostic
        for each $h \in H$ “at least one element of h is faulty”
```

6.3 Transition Based Passive Diagnostic Method

This method is used when only transitions from reachable to unreachable state are available to the Diagnostic System. The method below assumes that only one fault is happening for each polling cycle. The algorithm uses the following data:

- polling_period is an upper bound on the time that the Information Gathering System takes to query all the elements in the network;
- managed_elements is the set of network elements that are queried by the information gathering system;
- time() is a function that returns the current time.

```
For the first alarm that some S became unreachable
      tfirst = time()
      h = Path(S)
      faulty = {S}
for all alarms that S’ became unreachable and
   while time() < tfirst + polling_period
      C = Path(S’)
      h = h ∩ C
      faulty = faulty + {S’}
for all E in managed_elements- faulty
      C = Path(E)
      h = h - h ∩ C
print the diagnostic “at least on element of h is faulty”
```

The algorithm assumes that if there is no information about a network element after a polling cycle from the moment it received the first alarm, then that element is reachable. Thus, at the last loop, all elements that are not known to be faulty (because no unreachable alarm was received) are assumed to be reachable and the hypothesis is updated, using the Network Path Model.

7. The Global Architecture

7.1 Network Discovery System

An automatic or semi-automatic discovery system of the network configuration model can be implemented using a Management Network Platform. In general, a network management platform has a *discovery module* which is capable of constructing part of the network configuration model, but usually not with all the necessary information for a diagnosis system. This discovery system frequently collects information about Subnetwork Domain Level, and sometimes, about Equipment Level (equipment and interconnections).

In this work, we have developed a **Network Discovery System**. This prototype system runs on the Management Network Platform to collect network level information, previously discovered by the platform, and to interact with the agents in order to gather additional information necessary for the construction of the Configuration Model (Equipment Level, Network Element Level and Subnetwork Level).

The **Configuration Model Constructor** System uses the collected information of the Network Discovery System to construct the network Configuration Model. In this work, the Configuration Model will only contain the levels important to solve communication problems of a standard TCP/IP local area network, that is: the Equipment Level, the Network Element Level and the Subnetwork Level.

Because the Fault Diagnostic System could not reason about an inconsistent configuration model, it was necessary to implement a system to analyze the discovered network configuration, the **Configuration Diagnostic System**. Therefore, the Configuration Diagnosis System interacts with the *Configuration Model Constructor System* (see Figure 5) in order to find out possible configuration errors. Only when no errors are found, the Configuration Model becomes available to the Fault Diagnosis System.

The Configuration Diagnosis System contains several implemented functions to check the some characteristics like: if there are networks without elements; if the network is connected, if the network element is connected,; if there is the same IP number assigned to different elements; and if the routing tables are consistent with each other. In some cases is necessary to inform manually some equipment interconnections, when the relevant SNMP MIBs are not available in the agent.

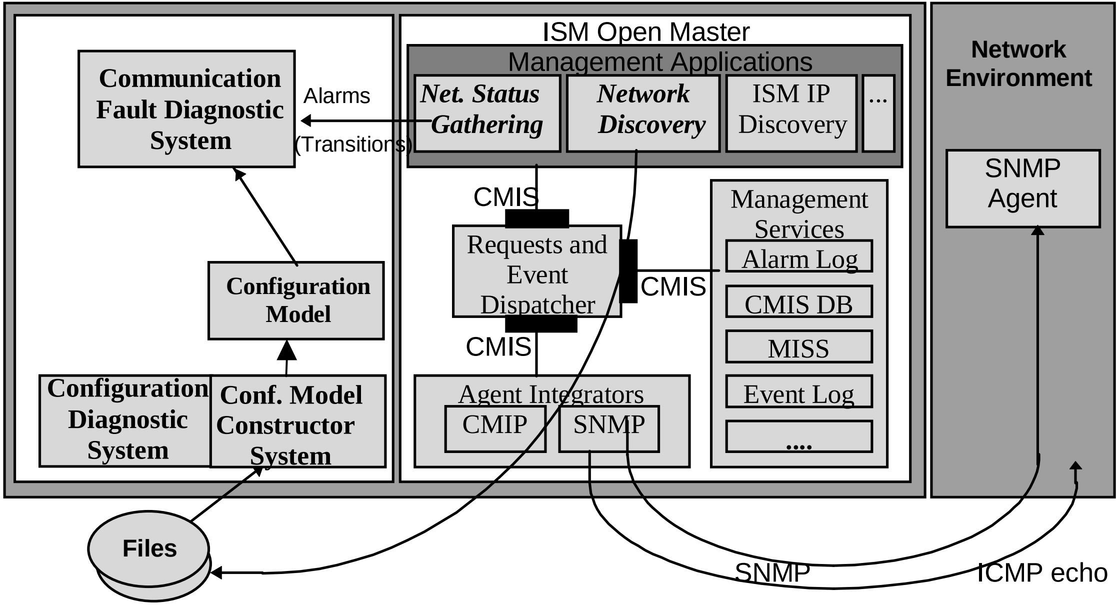

Figure 5. System Architecture.

Figure 5 presents the global architecture of the prototype system. The Communication Fault Diagnosis System and the Configuration Diagnosis System are processes whose programs are implemented in Prolog Language. The Communication Fault Diagnostic System receives the Configuration Model from the Network Discovery System and asynchronously alarms related to *communication status* from the Network Management Platform.

The Platform used is the *ISM OpenMaster* (*Integrated Management System* from Bull Company) that follows the OSI Standard. There are two implemented applications running over the ISM Network Management Platform: the *Network Status Gathering* and the *Network Discovery* applications. The *IP Discovery* is a native ISM application used to discover the equipments on the network. It collects some configuration data of each equipment and the subnetwork topology. This information is available on *CMIS DB* service. The *Network Discovery* System accesses the *CMIS DB* and creates the *Subnetwork Configuration Model*. For the construction of the *Equipment Level* and the *Network Element Level* models, it was necessary to inquire the agents about the Repeater MIB [RFC1515], the Bridge MIB [RFC1493], etc. This is done sending CMIP requests to the SNMP Agent Integrator. The SNMP Agent Integrator is an object manager that is responsible for converting CMIP requests from the Network Management Platform to SNMP requests.

The task of the *Network Status Gathering Application* is to receive all alarms and send only the relevant ones to the Diagnosis System in real time. In this case, the relevant alarms are those related to the communication state between the Network Management Platform and the equipment. The communication between the manager and equipment may be in three states: *SNMP up*, *IP up* and *unreachable*. The alarms are only generated on state transitions. Figure 6 presents the state diagram and the alarms that are generated when a transition occurs. This mode of alarm generation characterizes the system as a *Transition Based Passive Diagnostic System*.

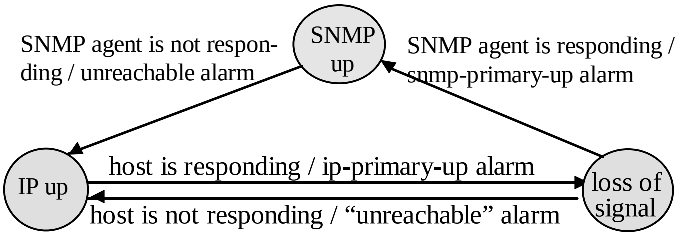
SNMP up
SNMP agent is not respon-ding / unreachable alarm
SNMP agent is responding / snmp-primary-up alarm
IP up
host is responding / ip-primary-up alarm
host is not responding / "unreachable" alarm
loss of signal

Figure 6. State Transitions Diagram of the communication between the manager and network devices.

The following table presents the alarms that are generated by the Agent Integrator Object Manager and that are used by the diagnosis system.

severity	alarm type	additional text
critical	communications	unreachable
major	communications	ip-primary-up
cleared	communications	snmp-primary-up

8. Conclusions

A prototype system to perform the communication fault diagnostic has been developed. The system can determine a network element (of a minimum set of network elements) that is causing a communication error, based on the unreachable alarms.

In order to construct the network configuration model automatically, a prototype of a subsystem for *Configuration Discovery* has been developed. Some management platforms only collect information at the Networks Level. The discovery system that was presented here uses information available in SNMP MIBs and can find out physical connections between network elements. It was also developed a subsystem to diagnose configuration errors which can detect and alert the network management team for configuration errors (inconsistencies between routing tables, for example) eventually causing other communication faults or performance bottlenecks. Both subsystems were tested on artificially constructed networks.

It is important to point out that as part of the process of developing the prototypes, we have got experience and insight on several issues that need to be addressed when developing diagnostic systems for networks, such as: **1)** the specification of an ontology for the task of network diagnosis; **2)** identification of a taxonomy of network faults and abnormal behavior, since for each class of fault, there is the need to develop a specific set of methods and models; **3)** identification of the possible modes of interaction between the diagnostic system and the network; **4)** specification of several diagnostic methods based on the different modes of interaction between the diagnostic system and the network; **5)** specification of many network models that are necessary to represent the network and the environment.; **6)** modeling some of the temporal aspects of network diagnosis.

One future work, is the addition of performance diagnosis of latency and bandwidth problems. For this, it will be necessary the addition of other models like, for instance, the causal model [22].

9. References

[1] Abu-Hanna, A. Multiple Domain Models in Diagnostic Reasoning. Amsterdam, 1994. 169p. **Thesis (Ph.D.)**, University of Amsterdam.

[2] Artola, E. S. e Tarouco, L. M. R. Um Sistema Especialista para Gerência Pró-Ativa Remota. In: SBRC - **Simpósio Brasileiro de Redes de Computadores**, 14^{o} , Fortaleza, 20-23 maio, 1996. ***Anais***. Fortaleza, SBC, 1996, p. 118-139.

[3] Benjamins, V. R. Problem Solving Methods for Diagnosis. Amsterdam, 1993. 172p. **Thesis (Ph.D.)**, University of Amsterdam.

[4] Bouloutas, A.; Calo, S.; Finkel, A. Alarm Correlation and Fault Identification in Communication Networks. **IEEE Transactions on Communications**, v.42, p.523-533, 1994.

[5] Breuker, J.; van de Velde, W.. ***CommonKADS Library for Expertise Modelling Reusable problem solving components***. **IOS Press**, Amsterdam, 1994.

[6] Chandrasekaran, B. Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design. **IEEE Expert**, vol. 1, no. 3, p. 23-30, Fall 1986.

[7] De Kleer, J. and Willians B. C.. Diagnosing Multiple Faults. Artificial Intelligence 32 (1987) 97 - 130.

[8] Deng, R. H.; Lazar, A.; Wang, W. A Probabilistic Approach to Fault Diagnosis in Linear Lightwave Networks. **IEEE JSAC**, v. 11, p. 1438-1449, 1993.

[9] Dreo, G.; Valta, R. Using master tickets as a storage for problem solving expertise. In: ISINM - International Symposium on Integrated Network Management, IV, ***Proceedings***, Chapman & Hall, p. 290, 1995.

[10] Frohlich, P.; Jobmann, K.; Nejdl, W.; Wietgrefe, H. Model-based alarm correlation in cellular phone networks. In: Fifth International Symposium on Modelling, Analysis and Simulation of Computers and Telecommunications Systems, 1997. ***Proceedings***, p. 197-204, 1997.

[11] Frontini, M.; Griffin, J.; Towers, S. A Knowledge-Based System for Fault Localization in Wide Area Networks. In: ISINM - International Symposium on Integrated Network Management, II, 1991. ***Proceedings***, Elsevier, North-Holland, p. 519-530, 1991.

[12] Katzela, I.; Schwartz, M. Schemes for Fault Identification in Communication Networks. ***IEEE/ACM Transactions on Networking***, Dec., 1995. http://www.ctr.columbia.edu/~irene/publications.html [obtained 01 Feb 1997].

[13] Kehl, W.; Hopfmuller, H.; Koussev, T., Newstead, M. Application of Model-Based Reasoning to the Maintenance of Telecommunication Networks. In: 5th International Conference, IEA/AIE. ***Proceedings***, Poderbon, Germany, June, 1992.

[14] Lemos, M. Um método de resolução de problema reusável para diagnóstico automático no domínio de gerenciamento de falhas em redes de comunicação. 16º Simpósio Brasileiro de Redes de Computadores - **SBRC98**. Page. 106-121. 1998

[15] Lewis, L. A case based reasoning approach to the resolution of faults in communication networks. In: ISINM - International Symposium on Integrated Network Management, III, 1993. ***Proceedings***, Elsevier, North-Holland, 1993.

[16] Brugnoni S., Bruno G., Manione R., Montariolo E., Paschettra E., and Sisto, L. (1993). An Expert System for Real Time Fault Diagnosis of the Italian Telecommunications Network. In: Third International Symposium on Integrated Network Management, San Francisco 18-23 April, 1993. The Netherlands, North Holland, 617-628.

[17] McDermott, J. Preliminary steps toward a taxonomy of problem-solving methods. In: Marcus, S. (editor), ***Automating Knowledge Acquisition for Expert Systems***, p. 225-255. Boston, Kluwer, 1988.

[18] Nunes, C. M. ***Um Discriminador Inteligente de Eventos de Rede para o ambiente CINEMA***. Porto Alegre, 1997, 143p. Dissertação (Mestrado) - CPGCC, UFRGS.

[19] Steels, L. *Components of expertise*. ***AI Magazine***, v. 11, p. 29-49, Summer 1990.

[20] Wietgrefe, H.; Tuchs, K. D.; Jobmann, K.; Carls, G.; Frohlich, P.; Nejdl, W.; Steinfeld, S. Using Neural Networks for Alarm Correlation in Cellular Phone Networks. ***In: IWANNT'97 - International Workshop on Applications of Neural Networks to Telecommunications***, Melbourne, Australia, June, 1997. http://www.kbs.uni-hannover.de/ [obtained 15 Jul. 1997].

[21] [Davis & Hamscher, 88] Model-based reasoning: Troubleshooting. In Shrobe, H. E., editor. ***Exploring Artificial Intelligence***, pages 297-346. San Mateo, California, Morgan Kaufmann.

[22] Lemos, M; Barros, L; Bernal, V; Wainer, J. Building Reusable Knowledge Models for the Communication Networks Domain. In Fourth Australian Workshop on Knowledge Acquisition, Sidney 5-6 December 1999.