Optimization of fault diagnosis based on the combination of Bayesian Networks and Case-Based Reasoning

Conference Paper · April 2012

DOI: 10.1109/NOMS.2012.6211970

5 authors, including:

Laurent Ciavaglia

Abdelghani Chibani
Université Paris-Est Créteil Val de Marne - Université Paris 12

Abdelhamid Mellouk
Université Paris-Est Créteil Val de Marne - Université Paris 12

Optimization of fault diagnosis based on the combination of Bayesian Networks and Case-Based Reasoning

Leila Bennacer, Laurent Ciavaglia
Alcatel-Lucent Bell Labs France,
Centre de Villarceaux,
Route de Villejust,
91620 Nozay, FRANCE
Email: {leila.bennacer, laurent.ciavaglia}
@alcatel-lucent.com

Abdelghani Chibani, Yacine Amirat,
Abdelhamid Mellouk
University of Paris-Est Creteil Val de Marne,
Image, Signal and Intelligent Systems Lab,
122 rue Paul Armangot,
94400 Vitry sur Seine, FRANCE
Email: {chibani, amirat, mellouk}@u-pec.fr

***Abstract*—Fault diagnosis is one of the most important tasks in fault management. The main objective of the fault management system is to detect and localize failures as soon as they occur to minimize their effects on the network performance and therefore on the service quality perceived by users. In this paper, we present a new hybrid approach that combines Bayesian Networks and Case-Based Reasoning to overcome the usual limits of fault diagnosis techniques and reduce human intervention in this process. The proposed mechanism allows identifying the root cause failure with a finer precision and high reliability while reducing the process computation time and taking into account the network dynamicity.**

Keywords: : Fault diagnosis, Root cause analysis, Case-Based Reasoning, Bayesian Network, Optimization, Message Passing inference.

I. Introduction

According to the functional model defined by the ITU-T specification [1], a network management system incorporates traditionally the following functions: Fault, Configuration, Accounting, Performance and Security (FCAPS). In this paper, we focus on the fault management process which is usually composed of the following steps [16]:

- Fault detection: consists in capturing network problems from the observed symptoms and triggering relevant alarms or events.
- Fault localization (or root cause analysis): is responsible for identifying one or more possible causes of a failure (root cause) by processing the observed symptoms.
- Testing: determines the actual faults by checking the alternatives provided by the fault localization.
- Fault recovery: consists of an intervention for fixing a problem and restoring the network operation.

Fault diagnosis covers essentially the fault localization and testing steps, and includes a serie of challenges such as maintaining up-to-date knowledge of the relationships among managed entities, distinguishing a single fault from multiple symptoms with many side effects, taking into account the dynamic addition of new entities in the network, etc. [6].

Although many approaches have been proposed to improve the process of fault diagnosis, there is still a need for the development of new solutions able to fulfill network operators' requirements and face real deployment constraints. Based on an analysis of current practices, requirements for operational networks, and pros/cons of existing fault-diagnosis techniques, our work emphasizes the following objectives: automation, speed, accuracy, and reliability.

This paper presents a new hybrid approach combining Case-Based Reasoning (CBR) [2] and Bayesian Networks (BN) [8]. According to literature [21], [11], Bayesian Networks are currently the most powerful and popular diagnosis method. However, the complexity of inference in Bayesian Networks increases exponentially with the number of nodes. Hence, this technique is not suitable for large scale systems including a large number of components such as current and future networks with hundreds or thousands of elements. To overcome this limitation, we propose a combined case-based and Bayesian reasoning approach to improve the BN inference. While keeping the advantages of BN technique, the resulting solution improves the degree of automation of the diagnosis process and requires less intervention of human expertise.

The remainder of the paper is organized as follows: the related work is reported in Section II. Section III presents briefly the desired properties of fault diagnosis. Section IV describes the proposed method while Section V analyzes and compares its performances to those of the classic BN technique. Finally, a conclusion and future directions are provided in Section VI.

II. Related Work

The theoretical framework of the fault diagnosis problem has been extensively investigated in literature. Most of the existing approaches are based on graph theory, stochastic analysis, or artificial intelligence techniques in order to embed intelligence in network elements and reduce human intervention. Next, we briefly present the most preeminent examples:

Most articles that deal with the diagnosis problem propose the use of Bayesian Networks [18], [21], [11]: a Bayesian Network (also called Belief network) is a Directed Acyclic Graph (DAG) whose nodes represent random variables and the edges denote existence of direct causal influences between the linked variables. The strengths of these influences are expressed by forward conditional probabilities. In the context of fault diagnosis, the random variables represent the occurrence of network events (such as buffer fill ratio, packet loss rate, etc.). Bayesian Networks are one of the most widely used approaches to address the root cause analysis problem [21], [11]. However, this technique is not deployed because the Bayesian inference is NP-hard problem [10].

Codebook approach consists of a set of events grouped into symptoms and problems to form a correlation matrix [22]. In the context of fault diagnosis, this approach is used to identify a problem among observed symptoms using coding theory. The codebook is created in terms of causality graph, which can be created automatically. The codebook approach has the advantages of speed, and resiliency to high symptom loss rate. However, a dynamic network environment may result in frequent updates of the codebook which could be costly in terms of computation time.

In [18], a Rule-Based System is proposed to improve the fault diagnosis process. Rule-Based Systems are used as a way to store and manage the knowledge to interpret information in a useful way. The rule-based methods are fast in processing well-defined fault events. However, frequent changes in network topologies may lead to the frequent updates of many rules. The main limitation of this approach comes from the fixed or static nature of the rules which apply only in predefined or specific contexts. This method is therefore not well-suitable for topology changes.

In [19], authors advocate the use of Markov Chains. A Markov Chain is considered as a countable sequence of random variables (X_n) with values in a countable space E. The distribution of $(X_1, X_2, X_3, \cdots)$ has the following property: $P(X_{n+1} = e_{n+1} | X_n = e_n, \cdots, X_0 = e_0) = P(X_{n+1} = e_{n+1} | X_n = e_n)$.
The authors of [19] claim that the Markov Chain is efficient since it does not impose any additional special requirements on the type of applied instance, and it appears to be fast based on the derived computational complexity. This technique has however some limitations due to the fact that the information about the occurrence of intermediate events in a fault-propagation chain is neglected. Also, this technique addresses only the isolation of a particular fault without identifying the sequences of events that could have probably originated as a result of this fault.

Neural Networks have also been suggested for the fault diagnosis [12], [13]. Neural Network are systems composed of interconnected nodes called neurons that try to mimic operation of a human brain. They are capable of learning and resilient to noise or inconsistencies in the input data. The disadvantage of Neural Networks is that they require long training periods and that their behavior outside their area of training is difficult to predict.

In [4], authors propose the treatment of causal dependencies between alarms and faults, using Petri Nets. Petri Nets technique is one of the most popular mathematical modeling languages for the description of distributed systems [5]. This technique is well known as a powerful model for concurrent systems, but, in our case, using the classic Petri Nets to model such a network in a context of fault diagnosis may lead to some drawbacks. Indeed, classic Petri Nets do not take into account the notion of time, which is very important in fault diagnosis to estimate the extent of the impact of the fault.

To sum up, numerous fault diagnosis techniques, with their own advantages and drawbacks, have been proposed and investigated. There is currently no ideal solution and fault diagnosis is still an active field of research with open challenges. Answering these challenges should improve the fault diagnosis process and be a first step in the adoption and deployment of carrier-grade diagnosis mechanisms. To characterize our approach within this context, we identify first the main properties and requirements from a network operator viewpoint. From the previous analysis, we define then a reduced set of objectives that our solution must meet.

III. Desired properties for fault diagnosis

Manageability (which consists in providing automated, adaptive, and autonomous to manage a network) and Diagnosability (proper root-cause detection and analysis) are two essential design goals of current and future networks [9]. In order to satisfy these objectives, and thus potentially overcome the limitations of past approaches, we have listed a series of key properties our solution shall meet:

- Process automation: it consists in reducing the human intervention and need of human expertise in the process of fault diagnosis.
- Dependency modeling: it consists in discovering and identifying the (hidden) relations between different entities constituting the network.
- Scalability: the fault diagnosis approach must support networks composed of hundreds to thousands of interconnected elements without impacting negatively its overall efficiency.
- Dynamicity: the addition (or removal) of elements in the network topology shall not impact the performance of the diagnosis solution. A network topology is subject to frequent changes and diagnosis solution must take into account this dynamicity and update dependencies automatically.
- Speed: the fault diagnosis approach must be fast enough to identify the root cause in a minimal time. Fault diagnosis is the most time-consuming step in the fault management process. Reducing the time to diagnose a problem can yield great performance improvement.

- Reliability: efficient solution for fault diagnosis must detect the correct root cause in a network containing hundreds or thousands of interrelated entities. This is achievable only if the dependencies between network entities have been properly identified.
- Accuracy: efficient solution for fault diagnosis must be able to precisely pinpoint the root cause among many alarms or symptoms and identify the problem(s) that need to be fixed in order to maintain the network service operation. Beside the identification of the root cause, the granularity of accuracy is an important indicator of the fault diagnosis overall efficiency.
- Genericity: the design of a reusable solution requiring only a small adaptation effort to be applicable to different network technologies.

IV. Description of the Proposed Approach

In this paper, we propose a new hybrid approach based on the combination of Bayesian Networks and Case-Based Reasoning techniques. The basic idea of our approach consists in the simplification, optimization, and automation of the diagnosis process to reduce the inherent complexity of Bayesian Network-based diagnosis [3].

In order to optimize the diagnosis process, we consider only a subset of the Bayesian Network structure, and, inside this subset, only the nodes where variations of the monitored parameters have been observed. This subset is identified thanks to Case-Based Reasoning [2]. CBR is also used for the learning phase of our approach. CBR learning allows improving the process efficiency over time by accelerating the identification and resolution of previously encountered pathological cases. The proposed approach is depicted in Fig .1.

A. Combination of Bayesian network and Case-Based Reasoning

As previously explained in Section II, and depending on the application domain, Bayesian Networks, as well as other methods such as Neural Networks, Petri Nets, Markov Chains, etc. can be considered as appropriate solutions to solve the fault diagnosis problem. Naturally, the choice of a specific method relies on different criteria, such as the complexity of the inference process, the cost and time to implement a solution.

The following features make Bayesian Networks, in many cases, a suitable technique:

- Knowledge representation: it is easy to maintain consistency and completeness in probabilistic knowledge bases [15].
- Conditional independence: independencies can be dealt explicitly. They can be calculated by an expert and encoded using graphical models. Every conditional independency embedded in the network can be recognized in linear time [15].
- Flexibility: Bayesian Networks allow using the same model to evaluate, predict, diagnose, and optimize decisions [17].

Despite its intrinsic advantages, the Bayesian Networks technique suffers from limitations that hinder performance and make this technique difficult to manipulate from a certain network size. As explained previously, the major qualities for an efficient fault diagnosis are the reduction of time needed for the identification of root cause, the reduction of complexity of the process and the accuracy of the root cause identification. Case-Based Reasoning [2] can advantageously complement and improve Bayesian Networks qualities and be used to reduce their complexity. Case-Based Reasoning is built on the notion of case and follows four main principles. In CBR, a case denotes a problem situation, a previously experienced situation, which has been captured and learned in a way that it can be reused in the solving of future problems. The Four principles of CBR are:

- Retrieve a similar case in the case database;
- Reuse the information of the similar case to solve the new problem;
- Revise the proposed solution;
- Retain the parts of the new experience likely to be useful for future problem solving.

Each of these steps has a specific role to improve the diagnosis process based on Bayesian Networks. The first principle "Retrieve"' is used as an additional learning phase compared to pure Bayesian Networks. To address the complexity of Bayesian Network inference, we propose to save the solutions found after each treatment in the form of solution-case. This allows reusing it when needed and avoids repeating the computation of inference, which is highly time- and processing power-consuming.

The principles "Reuse" and "Revise" allow the reuse and adaptation of the existing solution in the case database without repeating the entire process. The added-value of CBR in this phase is the simplification of the problem to a search in the case database. CBR is a cyclic and integrated process of solving a problem, learning from past experience, solving a new problem, etc.

When a problem occurs, if no solution can be found in the case database to a new problem-case faced, the corresponding problem is considered as a new case, and is processed using the inference process. After this process completed, if the new solution is valid, it is saved in the case database. This step corresponds to the "Retain" principle of CBR.

B. The steps of our approach

The proposed approach consists of four main steps: (1) Construction of the Bayesian Network, (2) Expressing a failure occurrence as a problem-case, (3) Optimizing the inference thanks to the optimization of Message Passing, and finally (4) Expressing and saving the outcome as a solution-case. Fig .1 illustrates how these steps are orchestrated by the combination of the two techniques.

Note that the CBR *Adapt the solution-case* and *Evaluate the new solution-case* steps (grey boxes) have not been investigated in this article and remain for future work.

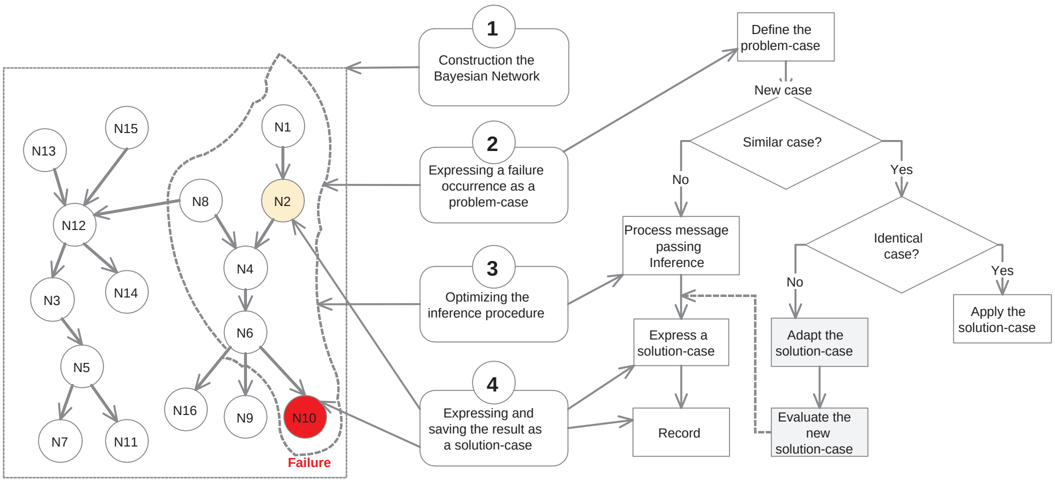

Fig. 1. Hybrid approach for fault diagnosis

1) Construction of the Bayesian network : For the construction of the relationships between nodes, the Chi-Square test [7] is applied instead of the heuristic algorithms generally used in Bayesian Networks. The Chi-Square test is used not only for the construction of dependencies but also to take into account new/removed nodes in the BN. The approach we propose aims at a rapid detection of the root cause while taking into account network topology changes, especially the addition or removal of nodes. When a new node is added, a Chi-Square test between this node and the others nodes is performed to find the new node position with respect to the initial Bayesian Network.

The Chi-Square test is always applied between two nodes of the Bayesian Network. The principle is to start with a hypothesis assumed true. In our case, we assume that two nodes are connected, which means that they sahre a dependency. Then, we prove the validity or not of this assumption. The approach is depicted in Algorithm 1:

Algorithm 1 Chi-Square test

if a new node is added to the Bayesian Network **then**
    Calculate $X^2 = \sum_{a=1}^{r_a} \sum_{b=1}^{r_b} \frac{(O_{ab}-T_{ab})^2}{T_{ab}}$
    if $X^2 < X^2_{theoretic}(df, 1-\alpha)$ **then**
        Graph[a,b]=1
    else
        Graph[a,b]=0
    end if
end if

With:
α : risk threshold;
df : $(r_a - 1)(r_b - 1)$;
r_a: number of states (variables) of node a;
r_b: number of states (variables) of node b;
O_{ab}: observed values obtained from the observations of the network (statistical values) such that O_{ab} are the observed frequencies having simultaneously A = a and B = b;

TABLE I
TABLE OF OBSERVED VALUES.

	Xa	Xb	
Xa	a	b	L1
Xb	c	d	L2
	C1	C2	N

T_{ab}: theoretical values obtained from the table of observed values such that:
$a' = \frac{C1 \times L1}{N}$, $b' = \frac{C2 \times L1}{N}$, $c' = \frac{C1 \times L2}{N}$, $d' = \frac{C2 \times L2}{N}$.

For the risk α and the degrees of freedom considered $df = (r_a - 1)(r_b - 1)$, if the Chi-Square value X^2 exceeds a threshold value called $X^2_{theoretic}$, the assumption is considered as invalid. Note that $X^2_{theoretic}$ value is given by a distribution table based on the value of α and the degrees of freedom $df = (r_a - 1)(r_b - 1)$. The commonly used value for the risk α is equal to 5 % (0.05).

As we shall see further, this approach allows taking into account the network dynamicity in the sense that the addition of new nodes does not impact negatively the network performance such as time of root cause detection, accuracy of root cause detection, etc.

Update of the Bayesian Network structure and especially of the probability tables (Chi-Square test) occurs not only when a new node is added to the network, but also anytime when an observed parameters changes. Thus, the learning of the network structure is quasi-continuous.

2) Expressing a failure occurrence as a problem-case: To simplify the process of identification of the root cause when a problem occurs, we represent the new problem as a problem-case. A problem-case is composed by the node where the failure has been detected, and the subset of nodes identified containing the new probability tables updated after detection of the new problem. The problem-case is identified in the Bayesian Network under study thanks to the concept of "case" in the CBR technique. It is represented as a subset of nodes on which our method will apply. When a problem occurs, instead of performing the inference on all the nodes of the Bayesian Networks, as it is usually the case, it is possible to run the inference process only on a subset of nodes showing variations. Indeed, the variation in the value of a node can result only from the value variation in the subset of its parent nodes. The identification of the problematic node and of parent nodes with variations allows determining the corresponding subset of nodes to be investigated for diagnosis. This subset is then analyzed to find the root cause.

During the exploration of the Bayesian Network under study, the nodes retrieval is done hop-by-hop starting from the node where the problem was detected. In our approach, instead of stopping at the nodes which variable values have not changed, we perform a verification two hops higher. We assume here that even if a node does not impact its direct child node (i.e., not changing the variables values); it can impact its second or third-degree descendant nodes.

We assume also that a node with an unchanged value cannot in any way affect another node. The algorithm 2 illustrates the process we propose to collect and filter the nodes to diagnose.

3) Optimizing the inference procedure: After the construction of the Bayesian Network (with the Chi-Square test), when a problem occurs and if no similar case can be found in the case database, we proceed to treat the case entirely. To do so, we propose the use of the "Message Passing" method to compute the inference with adjustment of the process to selected nodes (see algorithm 2).

Message Passing [20] is an algorithm mainly used for inference on graphical models, such as Bayesian Networks and Markov Chains. It calculates the marginal distribution for each unobserved node. The algorithm, proposed by J. Pearl [14], was initially formulated for tree structures, and then extended to polytrees. It is widely recognized as a powerful approximation algorithm for general graph problems that allow the calculation of the marginal distribution of each node.

In our context, assuming that the Bayesian Network is a Directed Acyclic Graph, the Message Passing algorithm is executed only on the subset of Bayesian Network identified. Upon the detection of a problem at a given node, the links with the nodes that are outside the subnet will be ignored, thus forming new leaves. This implies the construction of the Bayesian Network that is specific to the detected problem as presented in Fig. 2.

The principle of the adaptation of Message Passing inference can be summarized as follows:

Rach node communicates to its neighbors the information it collects, until all the nodes can update their marginal probability based on all the information received by the graph. This communication is done by transmitting messages between neighboring nodes through the edges between these nodes. The goal is that each node gets all the information it requires from its neighbors. During the execution of the Message Passing algorithm, each node X can be in one of the following states:

- Waiting for messages: let X be the number of new neighbors. Until receiving $N_x - 1$ messages, X remains in this waiting state.
- Computation of collected messages: after receiving $N_x - 1$ messages, X calculates the message to send to its only neighbor Y that sends him nothing. X is, in this case, in collection phase.
- Waiting for answer: X is awaiting a message from the last neighbor.
- Computation of messages distributions: once the last message received, X calculates:

Algorithm 2 Failure as a problem-case

```
for each metric(i) do
    metric(i).hop = 2;
end for
while new set of metrics ≠ ∅ do
    old set of metrics = new set of metrics;
    for each metric in new set of metrics do
        Send request (database, metric);
        Receive response (database, set of metrics);
    end for
    new set of metrics = new set of metrics ∪ set of metrics;
    Send get current value (new set of metrics, network);
    Receive (metrics current value);
    Send get normal value (new set of metrics, database);
    Receive (metrics normal value);
    for each metric(i) in new set of metrics do
        if metric(i).current value ≠ metric(i).normal value then
            set of metrics 1 = set of metrics 1 U metric(i);
            metric(i).hop=2;
        else
            Set of not changed metrics = Set of not changed metrics U metric(i);
            metric(i).hop= metric(i).hop -1;
            if metric(i).hop ≠ 0 then
                set of metrics 1 = set of metrics 1 U metric(i);
            end if
            for each metric(j) parent of metric (i) do
                metric(j).hop= metric(j).hop - 1;
            end for
        end if
    end for
    total set of metrics = total set of metrics ∪ set of metrics;
    new set of metrics = set of metrics 1- old set of metrics;
    set of metrics=∅;
end while
```

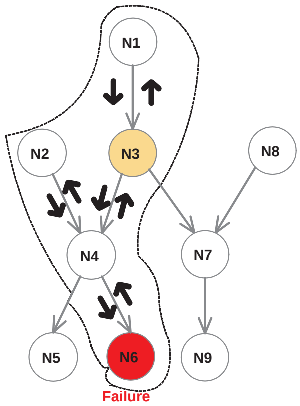

Fig. 2. Application of the Message Passing algorithm to the subset of the Bayesian Network

$$\lambda(X) = \prod_{1 \leq j \leq m} \lambda_{y_j(X)}$$

and

$$\pi(X) = \sum_{\mu \in D_\mu} P(X|Y) * \prod_{1 \leq i \leq t} \pi_x(\mu_i)$$

It sends then the $N_x - 1$ remaining messages. The latter are expressed as follows:

$$\lambda_x(\mu_i) = \sum_X \prod_j \lambda_{Y_j}(X) \sum_{U_k : k \neq i} P(X|\mu_1, \cdots, \mu_n) \prod_{k \neq i} \pi_X(\mu_k) \tag{1}$$

and

$$\pi_{y_j}(x) = \prod_{k \neq j} \lambda_{j_k}(x) \sum_{\mu_1, \cdots, \mu_n} P(X|\mu_1, \cdots, \mu_n) \prod_i \pi_x(\mu_i) \tag{2}$$

Note that $\lambda_x(\mu_i)$ messages are sent to parent nodes and $\pi_{y_j}(x)$ messages are transmitted to child nodes.
Once all the probabilities $P(X|e)$ have been computed by each node as $P(X|e) = \lambda(X) * \pi(X)$, we can do the comparison to identify among all the nodes the one that has generated the problem. The root cause is thus diagnosed.

- End: X is at rest state. The algorithm is completed.

As previously explained, the Message Passing algorithm is used in the inference process. By reducing the size of the network to process, the total computation time can be significantly reduced (see next section on evaluation results). Indeed, the exchanging of messages is the most time-consuming step as Message Passing is a process realized recursively by all nodes in the Bayesian Network during the inference process. To optimize the computation time, we limit the number of messages to only the nodes that are relevant for the problem observed in the system.

4) Expressing and saving the outcome as a solution-case : Once a problem has been identified and solved, the result is stored in the case database in the form of solution-case. This solution-case consists in the node where the problem has been detected, the identified root cause, and the subset of nodes selected with their probability table containing the variations due to this root cause occurence. This final step contributes to the learning phase of the proposed approach as it avoids duplicating the evaluation of the inference when a similar problem occurs in the future.

V. Evaluation

In this section, we evaluate and compare the performances of the combined CBR-BN approach to those of the classic BN technique. The evaluation results are organized according to five metrics, namely: Accuracy, Reliability, Speed, Scalability and Dynamicity.

A. Evaluation methodology and context

1) Generation of the Bayesian Network: To create and simulate a network topology, a set of nodes is generated using the BNJ[1] (Bayesian Network Tools in Java), an open-source suite software for BN. The generated network is evolutive since it is possible to dynamically add new nodes using the Chi-Square test presented previously. Each node is characterized by a marginal probability table, calculated from statistics collected from the observed network. Random data are introduced to fill the statistic tables.

2) Injecting faults and triggering symptoms : To generate a fault, a node in the network is randomly selected and a change in its marginal probability table is simulated. This root node directly affects one or more of his child nodes. One of the child nodes impacted by the previous changes is randomly chosen and, the Message Passing inference (according to CBR-BN approach) is used to find the root cause. Besides, a verification test is performed to determine whether or not the identified root cause corresponds to the original root node.

3) Simulation environment : All the performance evaluations have been realized on a basic, off-the-shelf computer (Intel Core 2 Duo T9600/2.8 GHz processor and 2GB of RAM). Obviously, results for the time to detect the root cause are dependent on the processing speed. Nevertheless, the figures presented allow drawing conclusion on the reduction of the complexity brought by our proposed solution and to evaluate the relative gain between the two approaches compared.

B. Evaluation results

1) Accuracy : Using the same statistics tables (the same marginal probability) for the two approaches CBR-BN and BN, we observe a significant difference in terms of accuracy. With BN, the root cause analysis accuracy decreases with an increasing size of the network under study, whereas in the the CBR-BN case, the level of accuracy remains constant.

Whatever the size of the original network, our proposed solution can precisely identify the root cause and with greater

[1] http://bnj.sourceforge.net/

accuracy than the BN technique. Note that we can get more than one root causes to an observed problem, which means that all identified root causes are real root causes. The observation of the results highlights that the BN technique is less accurate and the accuracy varies with the size of the network, e.g. from a set of 2 to 3 potential root cause nodes for a 10-nodes network, to a set of 6 to 8 nodes for a 500-nodes network. Another observation of the results shows that the root cause identified by the CBR-BN technique is always contained in the set of nodes identified by the BN approach.

Table II shows the results obtained for a network size ranging from 10 to 500 nodes. These first results demonstrate that our proposed solution provides satisfying precision in the root cause identification.

TABLE II
TEST OF ACCURACY.

Number of node in BN	BN approach	CBR-BN aproach
10 nodes	2 to 3	1
20 nodes	3 to 4	1
30 nodes	3 to 4	1
40 nodes	3 to 4	1
60 nodes	3 to 4	1
80 nodes	3 to 4	1
100 nodes	4 to 5	1
200 nodes	4 to 6	1
500 nodes	6 to 8	1

2) Reliability : In the previous section, the accuracy of our method has been evaluated. However, being accurate is not helping if the provided result is wrong. In this section, we evaluate the reliability of our solution to identify the correct root cause in a number of cases. We follow the same protocol as previously, where the reliability is demonstrated through the simulation of the root cause and the recording of the observed failures. To realize this test, we generated a network of 100 nodes, in which we randomly select a node as a root cause by changing its probabilistic table. Based on the observed faults, we used our approach to find back the root cause, and then check whether the root diagnosed by the CBR-BN technique is the right one and corresponds to the same the root cause previously recorded.

We modify the probabilistic table of the same root cause 100 times for each test case, and we followed the same verification protocol. The entire procedure is represented as a single test case in Fig. 3. Note that the test cases are initialized with a different root cause.

According to the dataset, the confidence interval, in 95% of cases, is approximately $(96.5, 98.7)$.

3) Speed: The time required to find the root cause is an important metric because it characterizes the overall diagnosis and fault management process performance. From the chart plotted in Fig. 4, we can learn two elements. Firstly, and as expected, the performance of the CBR-BN solution is significantly better than the one of the BN approach. The increase in the original network size has less impact on the detection time for the CBR-BN approach than for the BN approach (diverging curves). Secondly, compared to the

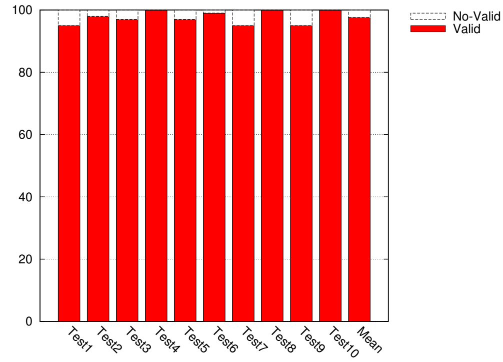

Fig. 3. Test of reliability

BN approach, the added-value of the CBR-BN approach is essentially the reduction of the complexity which appears in a significant reduction in the number of nodes involved in the diagnosis process. This reduction of the complexity translates directly into a far smaller time to indentify the root cause.

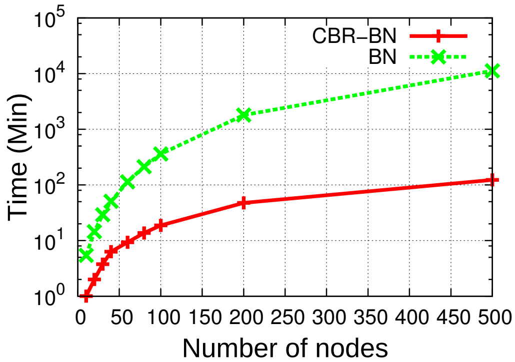

Fig. 4. Test of time detection of root cause

In order to illustrate this phenomenon, Table III details the number of nodes involved in the diagnosis process for the BN and CBR-BN cases. For networks greater than 40 nodes, the CBR-BN solution enables on average a reduction of the resulting network to a mere 1/10th of the original network size. This reduction in the size of the network to examine translates also into large gains in terms of speed as the CBR-BN outperforms the BN case, for all configurations, by 1 to 2 orders of magnitude.

TABLE III
REDUCTION OF NETWORK SIZE USING CBR-BN.

Number of node in BN	Size of the case for 9 tests									Median
10	2	2	3	2	3	3	3	3	2	3
20	2	3	4	4	4	3	5	3	4	4
30	4	4	3	2	3	4	4	4	5	4
40	4	5	6	3	4	5	3	5	5	5
50	5	5	4	6	5	5	5	4	6	5
80	7	8	5	7	5	6	5	7	7	7
100	10	9	10	11	10	11	11	10	10	10
200	23	21	21	18	19	16	21	20	22	21
500	46	54	49	53	59	42	54	49	62	53

4) Scalability: Using the Chi-square test, the simulation results demonstrate that no restriction is imposed on the size of the Bayesian Network under study, allowing an application of the proposed approach to very large topologies. This advantage allows reusing the CBR-BN approach whatever the network size and topology without detrimental effects on the performances.

5) Dynamicity: The results obtained in the previous tests (accuracy and root cause time detection) show that the removal or the addition of nodes in the network does not affect the performances of the CBR-BN approach. The latter ensures stable performances, despite network dynamicity in terms of topology change.

VI. Conclusion and future work

In this paper, we discussed a contribution to the area of fault diagnosis. We proposed a new approach based on a combination of graph theory technique "Bayesian Network" with another technique of artificial intelligence "Case Based reasoning" used to optimize the inference in Bayesian Networks. An extensive evaluation was conducted in simulation and showed the benefits of the proposed approach in comparison to the pure Bayesian Network approach according to five main metrics, namely: accuracy, reliability, speed, scalability and dynamicity. The solution is simple, flexible and scalable. It outperforms the traditional Bayesian Network method in all criteria bringing an increased speed with gains up to two orders of magnitude, combined advantageously with a higher accuracy and reliability. The reduction of the complexity enabled by the technique is also promising and will be the subject of future investigations in order to formally prove the level of complexity reduction attainable.

Even with the interests of the proposed method, there are still many possible improvements that we can bring. Indeed, despite we have proved the effectiveness of our solution to be generic, an application in real case(s) remains to be demonstrated. As an immediate objective, we intend to apply our approach on a case of fault-diagnosis for different types of communication services such as VoIP and VPN in a telecommunication network context. During the design of our approach, we have already tried to take in consideration all the specificities of these networks (topology change, scalability, etc.) and our results have proved theoretically that our approach could easily be applied, so the next step would be to prove this in a practical case, based on real data.

Acknowledgement

The research leading to these results has been performed within the UniverSelf project (www.univerself-project.eu) and received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n^{o}257513.

References

[1] The fcaps management framework: Itu-t rec. m. 3400.

[2] Agnar Aamodt and Enric Plaza. Case-based reasoning; foundational issues, methodological variations, and system approaches. *AI COMMUNICATIONS*, 7(1):39–59, 1994.

[3] Ashraf M. Abdelbar and Sandra M. Hedetniemi. Approximating maps for belief networks is np-hard and other theorems. *Artif. Intell.*, 102:21–38, June 1998.

[4] Armen Aghasaryan, Eric Fabre, Albert Benveniste, Renée Boubour, and Claude Jard. Fault detection and diagnosis in distributed systems: An approach by partially stochastic petri nets. *Discrete Event Dynamic Systems*, 8:203–231, 1998. 10.1023/A:1008241818642.

[5] René David and Hassane Alla. Petri nets for modeling of dynamic systems: a survey. *Automatica*, 30:175–202, February 1994.

[6] R. Dearden. Bayesian fault diagnosis: Common approaches and challenges. In *Cognitive Information Processing (CIP), 2010 2nd International Workshop on*, pages 215 –220, june 2010.

[7] Thomas Degris, Olivier Sigaud, and Pierre-Henri Wuillemin. Chi-square tests driven method for learning the structure of factored mdps. In *Proceedings of the 22nd conference on Uncertainty in Artificial Intelligence*, pages 122–129, Cambridge, MA, USA, 2006. Massachusetts Institute of Technology, AUAI Press.

[8] Eric. *Bayesian Networks of Dynamic Systems*. PhD thesis, Universitt'e de Rennes 1, March 2007.

[9] EC FIArch Group. Fundamental limitations of current internet and the path to future internet. Technical report, FIArch, 2011.

[10] Haipeng Guo and William Hsu. A survey of algorithms for real-time bayesian network inference. In *In In the joint AAAI-02/KDD-02/UAI-02 workshop on Real-Time Decision Support and Diagnosis Systems*, 2002.

[11] H. Kirsch and K. Kroschel. Applying bayesian networks to fault diagnosis. In *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, pages 895 –900 vol.2, aug 1994.

[12] Czeslaw T. Kowalski and Teresa Orlowska-Kowalska. Neural networks application for induction motor faults diagnosis. *Math. Comput. Simul.*, 63:435–448, November 2003.

[13] B. Li, M.-Y. Chow, Y. Tipsuwan, and J.C. Hung. Neural-network-based motor rolling bearing fault diagnosis. *Industrial Electronics, IEEE Transactions on*, 47(5):1060 –1069, oct 2000.

[14] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[15] Judea Pearl. Belief networks revisited. *Artif. Intell.*, 59(1-2):49–56, 1993.

[16] Gianluca Reali and Luca Monacelli. Definition and performance evaluation of a fault localization technique for an ngn ims network. *IEEE Transactions on Network and Service Management*, 6(2):122–136, 2009.

[17] Malgorzata Steinder and Adarshpal S. Sethi. Probabilistic fault localization in communication systems using belief networks. *IEEE/ACM Trans. Netw.*, 12:809–822, October 2004.

[18] Malgorzata Steinder and Adarshpal S. Sethi. A survey of fault localization techniques in computer networks. *Sci. Comput. Program.*, 53(2):165–194, 2004.

[19] Nikolay Tcholtchev, Agnieszka Betkowska Cavalcante, and Ranganai Chaparadza. Scalable markov chain based algorithm for fault-isolation in autonomic networks. In *GLOBECOM*, pages 1–6, 2010.

[20] John Winn and Christopher M. Bishop. Variational message passing. *J. Mach. Learn. Res.*, 6:661–694, 2005.

[21] Zhu Yongli, H. Limin, and Lu Jinling. Bayesian networks-based approach for power systems fault diagnosis. *Power Delivery, IEEE Transactions on*, 21(2):634 – 639, april 2006.

[22] Ming Yu, Wenjui Li, and Li jin W. Chung. A practical scheme for mpls fault monitoring and alarm correlation in backbone networks. *Computer Networks*, 50(16):3024 – 3042, 2006.