



Model-based probabilistic reasoning for self-diagnosis of telecommunication networks: application to a GPON-FTTH access network

Serge Romaric Tembo Mouafo, Sandrine Vaton, Jean-Luc Courant, Stephane Gosselin, Michel Beuvelot

► To cite this version:

Serge Romaric Tembo Mouafo, Sandrine Vaton, Jean-Luc Courant, Stephane Gosselin, Michel Beuvelot. Model-based probabilistic reasoning for self-diagnosis of telecommunication networks: application to a GPON-FTTH access network. *Journal of Network and Systems Management*, 2017, 25 (3), pp.558 - 590. 10.1007/s10922-016-9401-0 . hal-01533710

HAL Id: hal-01533710

<https://hal.science/hal-01533710>

Submitted on 6 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Model-based probabilistic reasoning for self-diagnosis of telecommunication networks: application to a GPON-FTTH access network

S.R. Tembo

Orange Labs, 2 Avenue Pierre Marzin, 22300, Lannion, France

S. Vaton

IMT Atlantique, 655 Avenue du Technopole, 29200, Brest, France

J.L. Courant, S. Gosselin, M. Beuvelot

Orange Labs, 2 Avenue Pierre Marzin, 22300, Lannion, France

*Published in Journal of Network and Systems Management,
volume 25, issue 3, july 2017*

Abstract

Carrying out self-diagnosis of telecommunication networks requires an understanding of the phenomenon of fault propagation on these networks. This understanding makes it possible to acquire relevant knowledge in order to automatically solve the problem of reverse fault propagation. Two main types of methods can be used to understand fault propagation in order to guess or approximate as much as possible the root causes of observed alarms. Expert systems formulate laws or rules that best describe the phenomenon. Artificial intelligence methods consider that a phenomenon is understood if it can be reproduced by modeling. We propose in this paper, a generic probabilistic modeling method which facilitates fault propagation modeling on large-scale telecommunication networks. A Bayesian network (BN) model of fault propagation on GPON-FTTH (Gigabit-capable Passive Optical Network-Fiber To The Home) access network is designed according to the generic model. GPON-FTTH network skills are used to build structure and approximatively determine parameters of the BN model so-called expert BN model of the GPON-FTTH network. This BN model is confronted with reality by carrying out self-diagnosis of real malfunctions encountered on a commercial GPON-FTTH network. Obtained self-diagnosis results are very satisfying and we show how and why these results of the probabilistic model are more consistent with

the behaviour of the GPON-FTTH network, and more reasonable on a representative sample of diagnosis cases, than a rule-based expert system. With the main goal to improve diagnostic performances of the BN model, we study and apply EM (Expectation Maximization) algorithm in order to automatically fine-tune parameters of the BN model from real data generated by a commercial GPON-FTTH network. We show that the new BN model with optimized parameters reasonably improves self-diagnosis previously carried out by the expert Bayesian network model of the GPON-FTTH access network.

1 Introduction

Telecommunication operators make significant efforts to provide better quality services to their subscribers. Telecommunication networks must be reliable and robust to guarantee high availability of services to customers. Network management has become a central issue for telecommunication operators, which have triggered significant research in order to automate as much as possible numerous complex operations of network management, like fault management. Fault diagnosis is a central aspect of network fault management [1].

The main goal of fault diagnosis is to locate as quickly as possible failures that degrade the quality of service provided to customers. Traditionally, fault diagnosis has been performed manually by an expert or a group of experts experienced in managing communication networks [1]. However, the development of telecommunication networks has increased the size and complexity of their architectures. Telecommunication networks have become large-scale complex distributed systems. A fault occurrence spreads, triggering other faults and alarms, which in turn trigger further faults and alarms. The consequence of both fault and alarm propagation is that a single root cause may result in a complex and distributed pattern of subsequent failures and their corresponding alarms [2]. This is especially true when multiple faults propagate simultaneously. Fault diagnosis has become too complex for humans, who can keep track of only a few hypotheses in their reasonings. Humans need a great deal of training to fully understand the fault propagation phenomenon in large-scale networks. Fault propagation is a complex phenomenon due to the dynamic, distributed and non-deterministic nature of telecommunication networks. A single fault may generate multiple alarms, and a single alarm may be triggered by several faults. Understanding fault propagation in order to automate fault diagnosis has become a critical issue

for operators. Understanding fault propagation is necessary to acquire relevant knowledge needed to perform self-diagnosis by automatically solving the problem of reverse fault propagation. This means using a configuration of alarms observed on the network to go back to the recent past in order to find their causal root explanations.

Two main types of methods can be used to understand fault propagation phenomenon in order to perform reverse fault propagation. Early self-diagnosis approaches of telecommunication networks, so-called expert systems, were rule-based. Expert systems [3] [4], encode specialized reasonings on narrow diagnostic tasks in computer applications. Artificial science methods consider that a phenomenon is understood if it can be reproduced by modeling and simulation, for example. In this category, we may distinguish model-based approaches [5], which develop reasonings based on an explicit representation of the network. The blind methods based on machine learning algorithms, like artificial neural networks [6] [7], and case-based reasoning [8], infer diagnosis based on past experiences, without network modeling.

In [9], we proposed an artificial method which combines the advantages of model-based approaches and machine learning approaches. This generic probabilistic method of network modeling integrates two fields: a decision field and an artificial learning field. The decision field is based on Bayesian network probabilistic reasoning [10] [11] in order to deal with the uncertainties of fault propagation phenomenon. The artificial learning [12] field is introduced and brings self-reconfiguration capabilities to the generic model in order to deal with the dynamic nature of telecommunication networks.

This paper explains how and why probabilistic model-based methods improve self-diagnosis of telecommunication networks by solving main drawbacks of expert systems. We propose a Bayesian network (BN) model of fault propagation on GPON-FTTH (Gigabit-capable Optical Network-Fiber To The Home) [13] [14] access network. With our Python implementation of this probabilistic model, we carry out self-diagnosis on a large-scale commercial GPON-FTTH network. Parameters of the BN model, approximately determined from GPON-FTTH network skills, are optimized by applying EM (Expectation Maximization) algorithm from real GPON-FTTH network data.

Section 2 of this paper presents a description and discussion of related

work on fault diagnosis, as well as the comparison between expert systems and probabilistic model-based methods. We also explain in this section why and how a probabilistic model-based approach improves self-diagnosis of telecommunication networks compared to expert systems. Section 3 recalls basic concepts on Bayesian network models. Section 4 presents our designed generic probabilistic model, which facilitates fault propagation modeling on large-scale telecommunication networks. In Section 5, we apply and implement this generic method to GPON-FTTH access network modeling. Section 5 also presents and analyzes self-diagnosis results of our implementation compared with those of a rule-based expert system. We study EM (Expectation Maximization) algorithm in Section 6 and apply it in Section 7 to automatically adjust parameters of the GPON-FTTH network model from real data generated by a commercial GPON-FTTH network. Section 7 also presents and analyses diagnosis results of the probabilistic model with optimized parameters comparatively to diagnosis results of the previous GPON-FTTH network model. We conclude and present future works in Section 8.

2 Expert diagnosis systems and model-based approaches

Early self-diagnosis approaches were called expert systems. An expert diagnosis system [3] [4] attempts to infer the cause of a problem from symptoms recognized in sensor data [2]. It is problem-solving software that embodies specialized reasonings on narrow diagnostic tasks, usually performed by a trained skilled human called an expert. The specialized reasonings can be formalized with rules, list of facts, logic predicates, etc. Inference engines are commonly based on forward-backward chaining algorithms. A review of diagnostic expert systems is available in [3] [4].

2.1 Rule-based expert systems

In the telecommunication industry, the most commonly used expert systems are those that use rules to represent specialized diagnostic knowledge, so-called rule-based expert systems. A rule may be equated to a matching between observed symptoms and their corresponding causes. A rule takes the form *IF* < *condition* > *THEN* < *decision* >. The condition may be evaluated directly if required data are available. Automated checks or tests on network components may also be necessary to evaluate the con-

dition. These tests increase the load of network components and generate additional network management traffic. If there are missing data, the condition cannot be evaluated and no decision can be taken. The decision can be the final diagnosis, i.e. the so-called conclusion, or the search for a more appropriate rule by using the rule inference engine. The inference engine has the ability to recognize input facts and infer output faults by searching for the most suitable rules to the recognized facts. Diagnosis computing follows a cycle of fact recognition and rule inference that ends when the most appropriate rule is found. If some facts cannot be recognized due to missing data, the diagnosis computing cycle is interrupted and the expert system does not produce a decision.

Fault propagation in telecommunication networks is naturally a dynamic, distributed and non-deterministic phenomenon. This means that it is very difficult for human experts to design a set of rules which covers all possible situations that may occur on the network. Humans can keep track of only a few hypotheses in their reasonings and need a long training period called an expertise period to fully understand fault propagation on a network segment. But despite these significant efforts to fully master the network, there are always unforeseen situations which require an extension of the expertise period. The set of rules designed by human experts can become obsolete if the new rules designed to cover an unforeseen situation require the modification of old rules. Rule-based expert systems suffer from this paradigm, referred to as the knowledge acquisition bottleneck that make them very difficult to maintain and unsuitable to carry out self-diagnosis of large-scale telecommunication networks. An expert system is a static and deterministic approach inappropriate to solving the non-deterministic problem of reverse fault propagation in large-scale telecommunication networks.

2.2 Model-based approaches

Contrary to rule-based expert systems, model-based approaches no longer encode specialized reasonings on narrow diagnostic tasks. Model-based approaches use specialized knowledge about the network to build an explicit, structured representation of network topology and network behavior. Model-based diagnostic approaches [15] [16] [17] [18] [19] [20] develop reasonings based on explicit, formal representation of network structure and network behavior. Network structure describes the network architecture. Network behavior describes the process of alarm propagation and alarm correlation [21]. Network structure and network behavior are then mod-

eled [5]. The obtained model is the support of reasoning algorithms which must be designed. Since reasonings are carried out on a model of the network, model-based approaches cover more situations than expert methods. They have the ability to deal with new problems or unforeseen situations, although their performance may degrade in these cases. Model-based approaches are easier to maintain than expert systems. The model can be designed in a modular or incremental fashion, facilitating updates as new knowledge about the network is acquired.

The term self-modeling is recently used in the literature to indicate a model-based approach whose the model is built automatically. In [2] a self-modeling method based on patterns describing in generic manner dependencies among resources used by an IMS (IP Multimedia Subsystem) service is proposed. A pattern is based on a Bayesian network, it is built offline, automatically located and instantiated online when a fault occurs in a given IMS service. In [22] a similar self-modeling method applied in the context of SDN (Software Defined Network) and NFV (Network Functions Virtualization) is proposed. It is based on templates which model SDN elements. An algorithm parses the network topology given by the SDN controller and a template based on a Bayesian network, is instantiated with eventually additional dependencies. A Bayesian network formalism is also used in [23], to model the propagation of a crosstalk attack in an optical network architecture. The purpose of this work is to bring and evaluate resiliency of optical network architectures under in-band crosstalk attacks. In [24], a fault propagation model based on petri network is developed, such that places on petri network represent fault and transitions represent dependencies between fault. Transitions are eventually labeled by the corresponding alarm pattern. The diagnosis amounts to compute the most likely path given a sequence of observed alarms. Note that in above examples, the model is graphic. But, the model can be analytic. In [25], analytics models and monitoring metrics are used to locate physical layer impairment in WDM (Wavelength Division Multiplex) optical networks.

A model-based approach seems natural when relationships between objects are graph-like and easy to obtain [1]. As we have seen in some above examples, the model can be probabilized in order to deal with uncertainties related to the non-deterministic nature of fault propagation.

However, it is quite difficult to build a model close enough to the structural and functional reality of the network, while maintaining a high enough

level of abstraction to make the model independent of the various engineering techniques implemented in telecommunication networks. In addition, the built model is reduced to a static image of the network and may become rapidly obsolete when the network changes. In [9], we have proposed a probabilistic generic model which embeds modularity and extensibility properties in order to facilitate self-reconfiguration of the model when the network changes.

2.3 Comparison between rule-based expert systems and a probabilistic model-based approaches

Table 1 summarizes the comparison between rule-based expert systems and probabilistic model-based approaches. We note that all the drawbacks of rule-based expert systems are solved by probabilistic model-based approaches.

Table 1: Comparison between rule-based expert systems (RES) and probabilistic model-based approaches (PMA).

Properties	RES	PMA
Scalability	No	Yes
Maintenance	Heavy	Easy
New problems	No	Yes
Uncertainty	No	Yes
Missing data	No	Yes
Ambiguous data	No	Yes
Scope	Narrow	Large
Learning	No	Possible

The brittleness of rule-based expert systems, i.e., their inability to deal with unforeseen situations, is a consequence of their case-by-case reasoning lacking generalized reasoning capabilities. In addition to their ability to deal with uncertainty and missing data, probabilistic model-based approaches have a very wide scope of reasoning that makes it possible to make global or generalized intelligent analyses of all possible situations that may occur on the network. Therefore, contrary to a probabilistic model, a rule-based expert system does not scale and fails when it encounters new problems. Expert systems do not have the ability to learn from experiences acquired from early diagnostic tasks. However, a model-based approach with no learning capability is a static image of the network. In [9], we proposed a

probabilistic generic model-based approach which embeds modularity and extensibility, two useful properties for easy self-reconfiguration of the initial network model. This generic model is based on Bayesian networks formalism. So we first recall this formalism.

3 Basic concepts on Bayesian networks

A Bayesian network [10] [12] [11] is a probabilistic directed acyclic graphical model which represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG). See Figure 1.

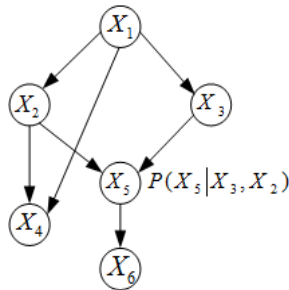


Figure 1: A simple Bayesian network of 6 random variables.

In a Bayesian network (BN) nodes represent random variables and edges represent dependencies. These random variables can be observations or measurements, latent variables or hypotheses for example. Each node is associated with a probability distributions that represent the distribution of the variable represented by the node, conditionally to the values of its parent nodes. These probabilities will be denoted as $\theta_{i,j,k} = \mathbb{P}(X_i = k | pa(X_i) = j)$ where this quantity stands for the probability that the value of node i is k given that the value of its set of parent nodes is j . A bayesian network encodes dependencies and independencies between a set of random variables. Indeed, any two variables in a BN model are dependent if one belong to the Markov blanket of the other, otherwise there are independent. Note that the Markov blanket of a variable in a BN is the union between the set of parents and children of this variable and the set of other parents of children of this variable.

There are different definitions of what a BN is. Let $B = (X_1, X_2, \dots, X_N)$ be the set of nodes of the BN. One possible definition is to state that the joint probability of random variables can be factored as the following product or

equivalently that the joint log-likelihood can be factored as a sum:

$$\begin{aligned}\mathbb{P}(B; \Theta) &= \prod_{i=1}^N \mathbb{P}(X_i | pa(X_i)) = \prod_{i=1}^N \theta_{i, pa(X_i), X_i} \\ L(B; \Theta) &= \log \mathbb{P}(B) = \sum_{i=1}^N \log \theta_{i, pa(X_i), X_i}\end{aligned}\quad (1)$$

Here the notation $\mathbb{P}(B; \Theta)$ (respectively $L(B; \Theta)$) makes explicit that the joint likelihood (respectively log-likelihood) depends on the set of parameters $\Theta = (\theta_{i,j,k})_{i,j,k}$. A BN embeds powerful and efficient computational capabilities. For example, it is possible to compute the conditional probability of any random variable or subset of random variables given an observation of one or many other variables, without performing heavy computations like marginalization. Indeed, efficient inference algorithms use the structure of the BN. Let's consider for example the fragment of a tree BN depicted by Figure 2.

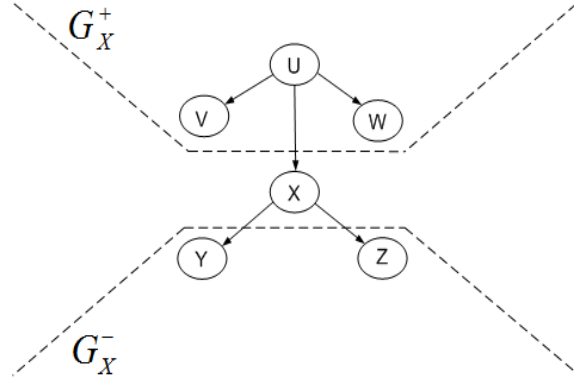


Figure 2: Segment of a tree Bayesian network.

Assume we want to compute the conditional probability of values of the variable X given some observed variables in the BN. The sum-product algorithm developed in [10], compute such conditional probability $BEL(x)$ given observations e_X^- collected in subgraph G_X^- containing the descendant nodes of X , and observations e_X^+ from the rest of the tree, i.e, the subgraph G_X^+ .

$$\begin{aligned}BEL(x) &= \mathbb{P}(x | e_X^+, e_X^-) = [\mathbb{P}(e_X^+, e_X^-)]^{-1} \mathbb{P}(e_X^+, x, e_X^-) \\ &= \alpha \mathbb{P}(e_X^- | e_X^+, x) \mathbb{P}(x | e_X^+) \\ &= \alpha \mathbb{P}(e_X^- | x) \mathbb{P}(x | e_X^+) \\ &= \alpha \lambda(x) \pi(x)\end{aligned}\quad (2)$$

where $\lambda(x) = \mathbb{P}(e_X^-|x)$, $\pi(x) = \mathbb{P}(x|e_X^+)$ and $\alpha = [\mathbb{P}(e_X^+, e_X^-)]^{-1}$ is normalizing constant. See [10] for detailed expressions of $\lambda(x)$ and $\pi(x)$. The sum-product algorithm use the structure of the BN since Equation 2 is verified because variables in G_X^+ are conditionnally independent from those in G_X^- given X . This conditional independence is true only for BN with no disoriented loop, i.e, a BN in witch any variable X does not belong to the set of its descendants, e.g. $X \notin G_X^-$.

For BN with disoriented loops like the BN depicted by Figure 1, a kindly simplification of such BNs, called junction tree, is proposed in [26] [27]. A junction tree is equivalent to a BN and inference algorithm on a junction tree is independent of the structure of the initial BN. This means that inference algorithm on a junction tree is very suitable for large and complex structures of BNs. Construction of a junction tree and inference on this structure is discussed in the next section which also describes our designed layered generic model for self-diagnosis of telecommunication networks.

4 A probabilistic generic model with self-reconfiguration capabilities

The generic model is based on a very simple principle: modeling the behavior of a telecommunication network amounts to modeling the behavior of the components that make it up. The generic model can be applied to any distributed system for self-diagnosis purposes.

4.1 Description of the generic model

Each network component E_i is modeled by two Directed Acyclic Graphs (DAG). One DAG, L_2N_i , models local fault propagation on the component in question (see Figure 3). Another DAG, L_1N_i , models distributed fault propagation between this component and components which are connected to it. Distributed fault propagation occurs between linked network components. The two DAGs L_2N_i and L_1N_i share some common variables. By separating local fault propagation modeling from distributed fault propagation modeling, the generic model also models network topology. In [9], we discussed the roles of layer 1 agents cL_1A , rL_1A , layer 2 agents cL_2A , rL_2A and layer 3 agents cL_3A , rL_3A (see Figure 3). We call layer 1 node, a DAG L_1N_i representing the network component E_i . We call layer 2 node, a variable of a DAG L_2N_i .

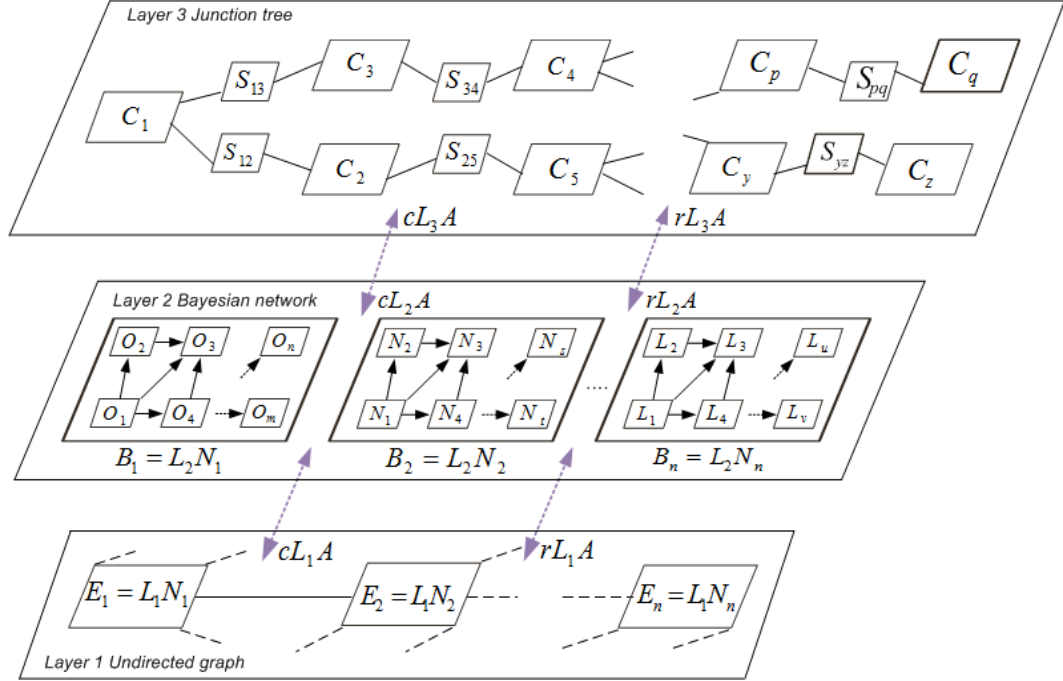


Figure 3: The 3-Layered Generic Model

The generic model has three layers. Layer 1 models the network topology as well as distributed fault propagation between linked network components. Any node E_i of this layer is represented by a Directed Acyclic Graph (DAG) noted L_1N_i which embeds distributed dependencies between the network component E_i and its neighbors. Note that $i \in \{1, \dots, n\}$ where n is the number of network components of the telecommunication network modeled. Distributed dependencies model fault propagation between linked network components. For example, the Figure 4 shows a very simple model based on the generic model. It is the model of a network segment of three components connected under a bus topology. In Figure 4, layer 1 node L_1N_2 contains distributed dependencies between E_1 and E_2 and between E_3 and E_2 .

Layer 2 models local fault propagation on each network component. For example, in Figure 4, DAGs L_2N_1 , L_2N_2 and L_2N_3 respectively model local fault propagation on network component E_1 , E_2 and E_3 . Layer 2 DAG

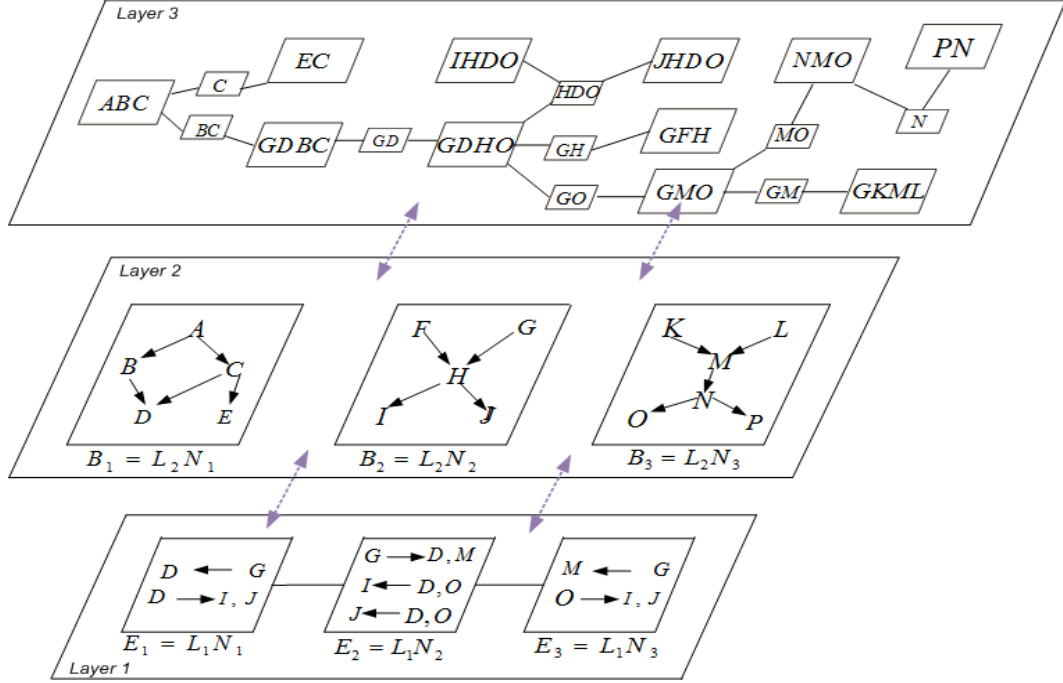


Figure 4: A simple example of the 3-Layered Generic Model.

L_2N_2 is connected to layer 2 DAGs L_2N_1 and L_2N_3 through layer 1 DAG L_1N_2 . The distributed nature of a telecommunication network introduces mutual dependencies between linked network components. These mutual dependencies lead to undirected loops between some variables of the BN model. For example in Figure 4, layer 2 DAG L_2N_1 contains the local undirected loop $[A, B, D, C, A]$ on network component E_1 . In Figure 4, the undirected loop $[C, D, G, H, I, D, C]$ is distributed between components E_1 and E_2 .

$$Gr = \bigcup_{i=1}^n [L_2N_i \cup L_1N_i] \quad (3)$$

Layer 3 is the junction tree representation [26] [27] [28] [11] of layer 1 and layer 2, i.e, the junction tree of the Bayesian network Gr defined by Equation 3. We call layer 3 node, a clique C_p of the junction tree. As every junction tree, layer 3 satisfies the running intersection property which ensures that, the intersection $C_p \cap C_q$ is a subset of every clique and separator of cliques on the path between C_p and C_q . For example, in Figure 4, $G = \{GDBC\} \cap \{GKML\}$ belongs to cliques $\{GDHO\}$, $\{GMO\}$ and

separators $\{GD\}$, $\{GO\}$ and $\{GM\}$ which form the path between cliques $\{GDBC\}$ and $\{GKML\}$.

Note that, layer 1 and layer 2 are useful for the self-reconfiguration field of the generic model, since they provide useful modularity and extensibility properties. Layer 3 is useful for the decision field of the generic model since inference can be easily done in this layer, regardless of network topology complexity at layer 1 and network behavior complexity at layer 2. See [9] for more details.

The fault propagation model on a large-scale telecommunication network of n components is decomposed in $2n$ sub DAGs, interconnected. This decomposition brings useful, easy self-reconfiguration properties to the model, modularity and extensibility [9]. Therefore, the fault propagation model of a large-scale telecommunication network can be build incrementally, network component by network component. The model can easily follow changes in network topology like adding or removing a network component. The important consequence is that we can start the network modeling by building a simple initial model from prior knowledge on a network segment. This initial model can be easily extended in order to take into account another network segment on which we recently acquired some knowledge about their behavior.

We introduced in [9] the specification of a reconfiguration protocol in order to automate the addition of a new network component into the initial network model or the removal of a network component from the current network model.

4.2 Construction of the junction tree

Consider the directed acyclic graph Gr defined by Equation 3. The first step to build a junction tree that forms layer 3, is to moralize DAG $Gr = (Ve, Ed)$ where Ve is the overall set of layer 2 nodes and Ed is the edges between them. The moral graph $Gm = (Ve, Em)$ of Gr is obtained by Equation 4 where Em is the set of edges of the moralized graph Gm .

$$(u, v) \in Em \iff [(u \rightarrow v) \in Ed] \vee [(v \rightarrow u) \in Ed] \vee [\exists w \in Ve \mid u, v \subset parent(w)] \quad (4)$$

The second step is to build a chordal graph $Gc = (Ve, Ec)$ from the

moralized graph Gm . We note ζ the set cycles of Gm , $ve(C)$ the set of vertices of a cycle $C \in \zeta$, $|ve(C)|$ the number of vertices of C and $ed(C)$ the set of edges of the cycle C . The chordal graph Gc is defined by Equation 5.

$$(u, v) \in Ec \iff [(u, v) \in Em] \vee \exists C \in \zeta \text{ with } |ve(C)| > 3 \text{ such that } [u, v \in ve(C) \wedge (u, v) \notin ed(C)] \quad (5)$$

Note that the quality of the triangulation largely determines the efficiency of the inference algorithm on junction tree. Triangulation is an optimization NP-hard problem. Indeed, an optimal triangulation is that which adds the smallest number of chords in the moralized graph, so-called minimum fill-in triangulation. In [29], a minimum fill-in triangulation algorithm is proposed.

The third step to build a junction tree is to compute the maximal weighted spanning tree of the triangulated graph Gc . The maximal spanning tree of the graph Gc is a subgraph that is a tree consisting of the subset of edges which together connect all cliques of Gc , while maximizing the total sum of weights on the edges. The weight is the size of the intersection between adjacent cliques, i.e, the number of layer 2 nodes shared by adjacent cliques. The maximal spanning tree of an undirected graph can be computed using the Kruskal algorithm.

4.3 Diagnostic computations of the generic model

Layer 3 is the junction tree representation of the large bayesian network Gr obtained by combining layer 1 and layer 2 as shown by Equation 3. This means that layer 3 is sufficient to compute diagnostic decisions using, for example, the well-known exact inference algorithm on a junction tree [26] [27]. In Figure 3, layer 3 nodes C_p , C_q , S_{pq} are respectively the cliques C_p and C_q of the junction tree and their common separator S_{pq} . A layer 3 node is a compound variable of some layer 2 nodes. At initialization, a clique C_p has the potential ϕ_{C_p} and a separator S_{pq} has the potential $\phi_{S_{pq}}$ as follows:

$$\begin{aligned} \phi_{C_p} &= \bigotimes_{X \in Layer2, X \in C_p, pa(X) \subset C_p \vee pa(X) = \emptyset} \mathbb{P}(X|pa(X)) \\ \phi_{S_{pq}} &= 1 \end{aligned} \quad (6)$$

We note $pa(X)$, the parent set of layer 2 node X . The potential ϕ_{C_p} of a layer 3 node C_p represents the joint conditional probability of the

layer 2 nodes that compose it. We recall that the product of distributions $\mathbb{P}(X_i|pa(X_i))$ of Equation 6 is not a matrix product but a tensor product of functions. Indeed, a distribution $\mathbb{P}(X_i|pa(X_i))$ is a function of variables $(X_i, pa(X_i))$.

We call evidence e , a configuration or a combinaison of values taken by a subset E of some layer 2 nodes. Let $E = \{X_1, X_2, \dots, X_t\}$ and $E \subset Layer2$, then $e = \{x_1, x_2, \dots, x_t\}$ such that x_i for $i \in \{1, \dots, t\}$ is a value taken by the layer 2 node $X_i \in E$.

The diagnostic decisions computed at layer 3 are based on evidence propagation on a junction tree. This algorithm starts by recomputing the initial potential of each observed clique or layer 3 node by setting observed layer 2 nodes belonging to this clique. A clique is observed if at least one layer 2 nodes of this clique is observed. After that, the evidence propagation algorithm essentially consists to update the potentials of any couple of adjacent cliques. Assume C_p, C_q to be neighboring layer 3 nodes with their common separator S_{pq} (see Figure 3). The algorithm updates the potentials of cliques C_p and C_q toward the separator S_{pq} .

$$\phi_{S_{pq}}^* = \sum_{C_p \setminus S_{pq}} \phi_{C_p}^* \quad (7)$$

$$\phi_{C_q}^* = \phi_{C_q} \frac{\phi_{S_{pq}}^*}{\phi_{S_{pq}}} \quad (8)$$

Assume for example that the potential of C_p has been already updated and the potential of C_q has not yet been updated, i.e, the propagation algorithm has already reached clique C_p but not yet the adjacent clique C_q . The evidence propagation algorithm proceeds in two steps. Firstly, the potential of the separator S_{pq} is updated by applying the marginalization operation of Equation 7. Secondly, the potential of the clique C_q is updated by the product of Equation 8. The notation $C_p \setminus S_{pq}$ represents the set of layer 2 nodes of clique C_p which does not belong to separator S_{pq} . We say that clique C_q absorbs evidence from C_p [21] or that clique C_p brings evidence to C_q .

Updating operations of potentials of cliques and separators are carried out in two recursive stages. The first stage called collect is initiated by collecting evidence from leave cliques to root cliques. This assumes that we arbitrarily choose an orientation on the junction tree by designating some

pending cliques as leave cliques of the tree and the other pending cliques are therefore root cliques of the tree. Root clique and leaves cliques respectively denote cliques with no predecessor and cliques with no successor in the junction tree according to orientation choosen. Collection of evidence to a clique C_p is done by collecting evidence to all children of C_p followed by absorption of evidences from each child. The second stage is initiated by distributing evidence from root cliques to leave cliques. Distribution of evidence from a clique consists to bring evidence to each child followed by distribution of evidence from the child.

Note that the update operations of the potentials of layer 3 nodes are carried out by an agent called computing Layer 3 Agent cL_3A . The cL_3A updates the potential of each layer 3 node when it receives evidence (observed layer 2 nodes) from its layer 2 counterpart called cL_2A . When cL_2A receives updated layer 3 node potentials, it computes the marginals (beliefs) of layer 2 nodes. These two agents may communicate using a simple mechanism like shared memory through an interface between layer 2 and layer 3, which we call the L2-L3 Interface (see Figure 5).

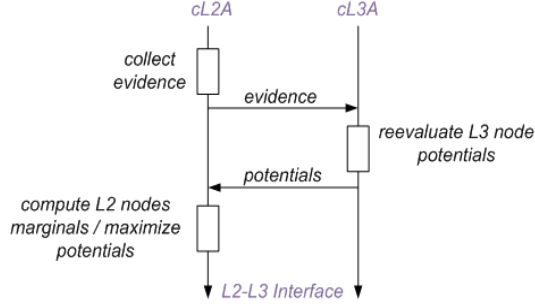


Figure 5: Communication between layer 2 and layer 3 for belief updating of layer 2 nodes

After updating the potential of all layer 3 nodes, the updated potential $\phi_{C_p}^*$ of the clique C_p is the joint probability of layer 2 nodes belonging to this clique and evidence e .

$$\phi_{C_p}^* = \mathbb{P}(C_p, e) \quad (9)$$

This means that the likelihood $\mathbb{P}(e)$ of evidence e and the conditional marginal probability $\mathbb{P}(X|e)$ of a layer 2 node X given e , can be computed

by applying Equation 10.

$$\mathbb{P}(e) = \sum_{C_p} \phi_{C_p}^* \quad \text{and} \quad \mathbb{P}(X|e) = \sum_{C_p \setminus X} \mathbb{P}(C_p|e) \quad (10)$$

Where $\mathbb{P}(C_p|e) = \phi_{C_p}^* / \mathbb{P}(e)$ is the joint probability of layer 2 nodes belonging to the clique C_i . The intersection property verified by a junction tree ensures coherence between updated potentials, i.e, the conditional marginal of a layer 2 node X given evidence e , is the same regardless of the clique C_p on which the marginalization operation is made. The most probable state of the layer 2 node X that is consistent with evidence, is the one that has the highest conditional marginal probability. The most propable state of layer 2 root nodes give us the diagnosis decision. Note that there is an alternative approach to finding the diagnosis without performing summation operations on updated potentials. With this approach, the diagnosis r^* , is computed from the most probable explanation w^* , of evidence as follows: $w^* = \bigcup_{C \in \text{Layer3}} w_C^*$ such that $\phi_C(w_C^*) = \max_{w_C} \phi_C(w_C)$, where w_C is a configuration of layer 2 nodes belonging to layer 3 node C and w_C^* maximizes the potential ϕ_C of C . Diagnosis r^* is the most probable configuration of layer 2 root nodes defined by $r^* = w^* \setminus i^*$, where i^* is the most probable configuration of non-root layer 2 nodes consistent with the evidence.

5 Case study: self-diagnosis of the GPON-FTTH access network

In this section, we apply the generic model to build an initial probabilistic model of the GPON-FTTH network. We force this model of self-diagnosis to confront reality by carrying out fault diagnosis on an operating GPON-FTTH network. The diagnostic results obtained with the probabilistic model are presented, analyzed and compared with those obtained by a rule-based expert system. Let us start with the presentation of the GPON-FTTH network and the probabilistic model-based approach for self-diagnosis of this network.

5.1 The GPON-FTTH access network

The GPON-FTTH access network has two main network components. The Optical Line Termination (*OLT*) is located on the operator side, and the Optical Network Termination (*ONT*) is located on the customer side, and

the *OLT* and *ONT* are connected through an Optical Distribution Network (*ODN*), which is the optical infrastructure made of fibers and passive components like splitters. A Passive Optical network (*PON*) is a point-to-multipoint link inside the *ODN*. A *PON* has a tree-like topology which connects an *OLT* with a maximum of 64 *ONT*s in our example (see Figure 6). Each *ONT* is connected to an *RG* (Residential Gateway) via an Ethernet link. Since there is no interaction between *PON*s in *ODN*, and all *PON*s have the same behavior, we modeled one single *PON*. This model can be replicated to any *PON* of a GPON-FTTH access network. All *ONT*s connected to the same *PON* temporally share the upstream optical channel of the *PON*. The downstream channel of the *PON* is a secured broadcasting channel.

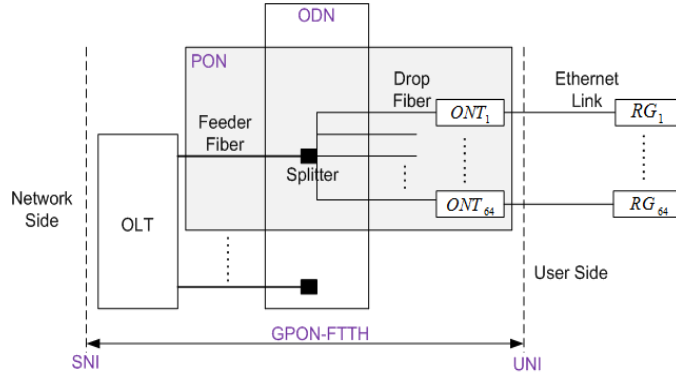


Figure 6: GPON-FTTH network architecture

Figure 7.A depicts the model of *OLT* component for self-diagnosis purpose. As designed by the generic model, this component is modeled by two DAGs. One DAG called layer 2 DAG for local fault propagation and a layer 1 DAG for distributed fault propagation between *OLT* and *ONT* components. The two DAGs are connected since they have some common variables. Figure 7.B depicts the model of *ONT* component. The two models are connected since they share common distributed dependencies via layer 1 (see Figure 7). Figure 7 presents the application of the generic model for modeling the topology and behavior of a *PON* of the GPON-FTTH access network. The obtained model has two layer 1 nodes, i.e., the components *OLT* and *ONT*. Some variables called layer 2 nodes are vectors in order to respect the tree-like topology of a *PON*. Indeed such layer 2 nodes represent a particular value (for example an alarm, a counter or a scalar parameter) at the different *ONT*s of the *PON*. Each element i of the vector refers to *ONT*

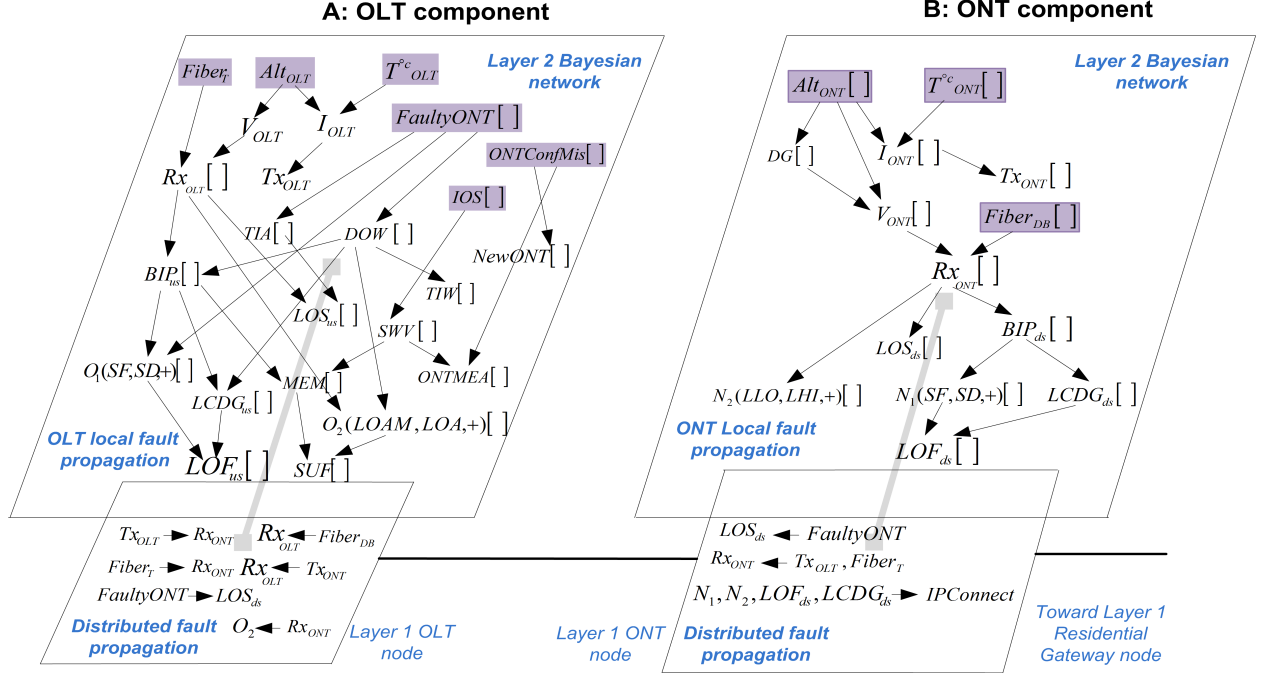


Figure 7: The GPON-FTTH model based on the generic model.

number i in the *PON*. For reasons of simplicity, layer 3 is not depicted. We have three types of layer 2 nodes on Figure 7: faults or root causes, intermediate faults and alarms. The root causes are highlighted in Figure 7.

The transport optical fiber of the *PON* denoted by $Fiber_T$ can take three states. The *OK* state means that there is no transmission anomaly on this fiber. The *AT* and *BR* states mean respectively that the fiber is experiencing high attenuation or that the fiber is broken. The temperature of *OLT* denoted by T_{OLT}^c , is a continuous variable that we discretize. The power supply of *OLT* denoted by Alt_{OLT} may or may not be faulty. The node *FaultyONT* denotes an *ONT* which transmits an upstream signal outside of its granted time slot, which may conflict with data sent by other *ONTs* on the *PON* and cause data disruption for a random set of *ONTs*, making the *PON* unusable. A *FaultyONT* can cause a Drift of Windows *DOW*. The component *OLT* raises a $DOW[i]$ alarm when an $ONT[i]$ transmits a signal beyond the time slot allocated to it. See ITU-T G984.3 [13] [14] for more details. We note $i \in \{1, \dots, 64\}$ the position of this *ONT* on the *PON*.

The Software Version $SWV[i]$ alarm means that there is an incompatibility between the Image Operating System (IOS) of $ONT[i]$ and those of the OLT . The node $ONTConfMis$ (ONT Configuration Mistake) denotes a configuration error during ONT provisioning. The OLT transmitted power Tx_{OLT} is regulated by the bias current I_{OLT} . This leads to the local dependency $I_{OLT} \rightarrow Tx_{OLT}$. The OLT received power $Rx_{OLT}[i]$ from an ONT_i depends on the OLT voltage V_{OLT} and the state of the feeder fiber $Fiber_T$. The OLT received power of ONT_i also depends on the state of the drop fiber denoted by $Fiber_{DB}[i]$ and the transmitted power of this ONT_i denoted by $Tx_{ONT}[i]$. Note two local dependencies ($Fiber_T \rightarrow Rx_{OLT}$, $V_{OLT} \rightarrow Rx_{OLT}$) and two distributed dependencies ($Fiber_{DB} \rightarrow Rx_{OLT}$, $Tx_{ONT} \rightarrow Rx_{OLT}$). Note in Figure 5 that the distributed dependencies are part of the edges of the layer 1 OLT , node which is a DAG as designed in the generic model.

The Bit Interleaving Parity denoted by $BIPus[i]$ depends on the Bit Error Rate BER of an upstream data transmission between an ONT_i and OLT . A poor upstream signal reception can cause bit errors, leading to the local dependency $Rx_{OLT} \rightarrow BIPus$. Upstream transmission bit errors impact the quality of the signal received by OLT , which may raise some alarms related to signal quality like SD (Signal Degraded), SF (Signal Fail), $LCDG$ (Loss of GEM Channel Delineation) and MEM (Message Error Message). See ITU-T G984.3 [13] recommendation for more details.

When the received power $Rx_{ONT}[i]$ of an ONT_i is less than a preconfigured minimum threshold, this ONT raises the (Level Low) $LLO[i]$ alarm. The (Level High) $LHI[i]$ alarm is raised by ONT_i when $Rx_{ONT}[i]$ is greater than a preconfigured maximum threshold. For simplicity and because the $LEVELLO$ alarm denoted by LLO and the $LEVELHI$ alarm denoted by LHI can not be observed simultaneously, we have considered them to be states of the layer 2 node called $N_2(LLO, LHI, +)$. The state denoted by $+$ means that there is no LLO or LHI alarm observed. The received power $Rx_{ONT}[i]$ depends on voltage $V_{ONT}[i]$, the state of drop fiber $Fiber_{DB}[i]$, the state of feeder fiber $Fiber_T$ of the PON and the transmitted power Tx_{OLT} of OLT . Note the local dependencies $V_{ONT} \rightarrow Rx_{ONT}$, $Fiber_{DB} \rightarrow Rx_{ONT}$ and the distributed dependencies $Tx_{OLT} \rightarrow Rx_{ONT}$, $Fiber_T \rightarrow Rx_{ONT}$.

Now, suppose we need to extend this model by adding another layer 1 node: a Residential Gateway RG node, for example (see Figure 6). An

RG is a home network component that provides services to customers. It is not a GPON-FTTH network component, but adding it to the model will enable the model to correlate the faults and alarms of the GPON-FTTH network with customer service malfunctions. For example, in Figure 5, we can make such a correlation with the distributed dependency $N_1, N_2, LOF_{ds}, LCDG_{ds} \rightarrow IPConnect$, where *IPConnect* denotes the Internet access service provided by an *RG*. To add the layer 1 *RG* node, we will need only to specify and quantify the uncertainties of its local dependencies and distributed dependencies with the layer 1 *ONT* node, which already exists in the model.

5.2 Parameters estimation of the GPON-FTTH network model

In the last subsection, we have showed how we can use skills about GPON-FTTH network to design a fault propagation model of this network. Here, we also show how these skills can be used to turn this model into a bayesian network by approximatively determining the parameters of this model, i.e, a probability distribution of each layer 2 node conditionally to its parents, which aims to quantify uncertainties on dependencies. Obviously, it is more prattice to adjust the parameters of a statistical model by applying machine learning techniques from a dataset. But, our choice to use expert knowledge about GPON-FTTH network, in order to find an estimation of parameters, is motivated by the fact that machine learning algorithm for parameters estimation of a bayesian network model is based on maximum likelihood principle of the dataset. This algorihm requires an initial value of parameters vector to perform estimation from incomplet dataset as it is often the case for a telecommunication network dataset. This initial value should not be aberrant, it must be quite realistic to hopefully avoid problems related to local optimum of the likelihood of data. Therefore parameters approximatively determined with GPON-FTTH network skills in this subsection, so-called expert parameters can later serve as initialization point to an algorithm such that EM (Expectation Maximization) [30] [31] which can therefore adjust them automatically from GPON-FTTH network data.

Consider a layer 2 node X_i of the GPON-FTTH model depicted by Figure 7. We note $\theta_{i,j,k} = \mathbb{P}(X_i = k | pa(X_i) = j)$ the probability that the variable X_i takes the value k when the combinaison of values taken by its parents is j . We propose to approximatively determine the conditional probability distribution $\theta_i = \mathbb{P}(X_i | pa(X_i))$, based on GPON-FTTH network skills. The idea is not precisely to find proportions $\theta_{i,j,k}$, but rather to

develop a structured reasoning consistent with the behavior of the GPON-FTTH network. We recall in the words of Glenn Shafer (a pioneer of the theory of probability applied to artificial intelligence) that "Probability is not really about numbers; it is about the structure of reasoning". For our problem, we search two orders of magnitude. An order of magnitude between proportions of the vector $\theta_{i,j,\bullet}$, for all possible values of k , with the constraints $\sum_k \theta_{i,j,k} = 1$ and $\theta_{i,j,k} \in [0, 1]$. An order of magnitude between the probabilities of the vector $\theta_{i,\bullet,k}$, for all possible values of j . In order words, we search an order of magnitude between elements of each lign and an order of magnitude between elements of each column of the 2-dimensions matrix $\theta_i = \mathbb{P}(X_i|pa(X_i))$. We illustrate with some examples below how such magnitude orders can be determined based on GPON-FTTH network skills.

The first exemple concerns the estimation of conditional probability distribution $\mathbb{P}(\mathbf{LOF}_{\text{us}}|\mathbf{O}_1, \mathbf{LCDG}_{\text{us}})$. Table 2 represents this distribution. We note $+lof$, the event of loss of data frame that is manifested by the observation of LOF (Loss Of Frame) alarm. $\neg lof$ is the complementary of $+lof$ event. On the upstream channel between OLT and one ONT, it is more probable that the loss of a data frame ($+lof$) occurs when an error of delineation of frames ($+lcd$, loss of gem channel delineation) transmitted by this ONT occurs, than when OLT carry out frames delineation without error ($\neg lcd$). This technical knowledge about GPON-FTTH network is represented by constraint 11 in which the event $+e$ corresponds $+lcd$ and the complementary $\neg e$ of $+e$ corresponds to $\neg lcd$, (see Table 2). Numerical values in this table are consistent with constraints 11, 12 and 13.

$$\theta_{p,q}^{-e} \leq \theta_{p,q}^{+e} \quad \text{for } p = 1, q \in \{1, 2, 3\} \quad (11)$$

Futhermore, since transmission errors rate is more important when SF (Signal Fail) alarm is observed than when SD (Signal Degraded) alarm is observed, then it is more likely that loss of frames occurs when signal degradation level is high (SF) than when it is medium (SD). We represent this information by constraint 12.

$$\begin{aligned} \theta_{p,q}^{+e} &\leq \theta_{p,q+1}^{+e} \\ \theta_{p,q}^{-e} &\leq \theta_{p,q+1}^{-e} \quad \text{for } p = 1, q \in \{1, 2\} \end{aligned} \quad (12)$$

Finally, we can say that it is more likely that a loss of frames occurs on the upstream channel when its two potential causes occurred, i.e, the

observation of a high or medium signal degradation via SF_{us} alarm or SD_{us} alarm, and the observation of a frames delineation error via $LCDG_{us}$, than when only one of two potential causes occurred. So, we can also consider the constraint 13 (see Table 2).

$$\theta_{1,2}^{-e} \leq \theta_{1,3}^{+e} \quad \text{and} \quad \theta_{1,3}^{-e} \leq \theta_{1,2}^{+e} \quad (13)$$

Table 2: The 2-Dimensions matrix that represents the conditional probability distribution $\mathbb{P}(LOF_{us}|O_1, LCDG_{us})$.

O_1	$LCDG_{us}$	$+lof$	$\neg lof$
$+$	$+lcd$	$\theta_{1,1}^{+lcd} \approx 0.700$	$1 - \theta_{1,1}^{+lcd}$
SD		$\theta_{1,2}^{+lcd} \approx 0.800$	$1 - \theta_{1,2}^{+lcd}$
SF		$\theta_{1,3}^{+lcd} \approx 0.800$	$1 - \theta_{1,3}^{+lcd}$
$+$	$\neg lcd$	$\theta_{1,1}^{-lcd} \approx 0.100$	$1 - \theta_{1,1}^{-lcd}$
SD		$\theta_{1,2}^{-lcd} \approx 0.700$	$1 - \theta_{1,2}^{-lcd}$
SF		$\theta_{1,3}^{-lcd} \approx 0.700$	$1 - \theta_{1,3}^{-lcd}$

The conditional probabilities of the distribution $\mathbb{P}(LOF_{us}|O_1, LCDG_{us})$ must satisfy the three constraints formalized by inequations 11, 12 and 13. In other words, these three constraints formalize GPON-FTTH networks skills encoded in the distribution $\mathbb{P}(LOF_{us}|O_1, LCDG_{us})$. Table 2 gives an approximative value of each conditional probability consistent with order of magnitude defined by these three constraints. The second example concerns the estimation of the conditional probability distribution $\mathbb{P}(LCDG_{us}|BIP_{us}, DOW)$. Table 3 shows how this distribution is approximatively determined based on constraints 11, 12 and 13 on which event $+e$ corresponds to event $+dow$ and the complementary $\neg e$ of $+e$ corresponds to $\neg dow$. The first constraint expresses that, for any transmission errors rate, it is more probable that an error of frames delineation occurs ($+lcd$ event) when a drift of slot transmission time is observed ($+dow$ event), than when it is not the case ($\neg dow$ event). The second constraint 12 formalizes that, regardless of $+dow$ event, the probability that an error of frames delineation occurs, increases with transmission errors rate BIP_{us} . According to the third constraint 13, it is more likely that an error of frames delineation occurs on the upstream channel when its two potential causes occurred, i.e, the observation of a drift of windown alarm, and a high or medium transmission errors rate, than when only one of two potential causes occurred.

Table 3: The 2-Dimensions matrix that represents the conditional probability distribution $\mathbb{P}(LCDG_{us}|BIP_{us}, DOW)$.

BIP_{us}	DOW	$+lcd$	$\neg lcd$
$+$	$+dow$	$\theta_{1,1}^{+dow} \approx 0.100$	$1 - \theta_{1,1}^{+dow}$
low		$\theta_{1,2}^{+dow} \approx 0.800$	$1 - \theta_{1,2}^{+dow}$
$high$		$\theta_{1,3}^{+dow} \approx 0.900$	$1 - \theta_{1,3}^{+dow}$
$+$	$\neg dow$	$\theta_{1,1}^{-dow} \approx 0.001$	$1 - \theta_{1,1}^{-dow}$
low		$\theta_{1,2}^{-dow} \approx 0.700$	$1 - \theta_{1,2}^{-dow}$
$high$		$\theta_{1,3}^{-dow} \approx 0.700$	$1 - \theta_{1,3}^{-dow}$

A similar probabilistic reasoning to those developed in Tables 2, 3 and constraints 11, 12 and 13 is used to approximatively determine the distribution of each variable of the GPON-FTTH network model depicted by Figure 7. These distributions turn the directed acyclic graph Gr defined by Equation 3 into a bayesian network (BN). Since the structure of the bayesian network Gr is designed by skilled humans on GPON-FTTH network, Gr is a causal bayesian network. Indeed, for humans, a simple and intelligible way to represent causes and effects relationships describing fault propagation, is a causal dependency graph. However if the structure had been learned by a machine learning algorithm from a data sample, arrow between two variables will not necessarily express causality but rather correlation between them.

We have used this bayesian network (BN) model to perform self-diagnosis of the GPON-FTTH network. In order to validate and assess performances of self-diagnosis with the BN model, we have used two different approaches. A first approach described in [9] was to set up a physical testbed with a PON with two ONTs. Different faults were emulated, and alarms as well as counters were collected. The diagnosis of the root cause of alarms was performed with the BN approach. Seven usual fault scenarios were considered. Diagnosis results were inspected manually in order to assess their reliability. This demonstrated that self-diagnosis based on a BN model was a reliable and promising approach. In a second phase, the BN model is confronted with reality by carrying out self-diagnosis on an operating GPON-FTTH network.

5.3 Self-diagnosis results on an operating GPON-FTTH access network

We present and analyze in this section the network fault diagnosis results carried out by our application and implementation of the generic model to perform self-diagnosis of a GPON-FTTH network. The experiments were performed on an operating GPON-FTTH network on which we have considered different PONs. For each PON considered, we call the *ONT* being tested, the first *ONT* of this PON, i.e., ONT_1 . The other *ONTs* connected to this PON are called the neighbors of ONT_1 . We collected the alarms raised by the operating PON network components, i.e., *OLT* and *ONTs*. We also read, if available, the values of counters BIP_{us} , BIP_{ds} and scalar parameters Rx_{OLT} , Rx_{ONT} , Tx_{OLT} , Tx_{ONT} , voltages V_{OLT} , V_{ONT} , bias current I_{OLT} , I_{ONT} , temperatures T_{OLT}^c , T_{ONT}^c of *OLT* and *ONTs* connected to the considered *PON*.

A database of 10611 real diagnosis cases collected by Orange on a commercial GPON-FTTH network in july-august 2015 was analysed. Two tools are compared: PANDA, the self-diagnosis tool based on the BN approach described in this paper, and DELC, a self-diagnosis tool based on deterministic decision rules. DELC is currently used to diagnose faults in the operational network. DELC is based on Drools, a business rules management system solution developed by the JBoss community, that provides a core business rules engine. We present and analyze quantified self-diagnosis results of a GPON-FTTH network carried out separately by DELC rule-based expert system and our probabilistic model-based tool PANDA. A 2-dimension confusion matrix depicted by Table 4 is used to compare the results obtained with each of the two methods.

In Table 4, each row of the confusion matrix is labelled by a diagnosis result obtained with the expert system. Each column is labelled by a diagnosis result obtained with the probabilistic model-based approach. Although the probabilistic model simultaneously carries out the diagnosis of all *ONTs* connected to the PON being considered, we have limited the comparison to the diagnosis of only one *ONT*. That is because the expert system outputs only the diagnosis of one *ONT*, namely the *ONT* under test, i.e., the *ONT* named ONT_1 in our PON model. The comparison is made on 10611 diagnosis cases. Observations of each case are collected from an operating GPON-FTTH network. For each case, the expert system and the probabilistic model separately perform a diagnosis based on the same observations.

Table 4: A 2-dimension confusion matrix between the results of the rule-based expert system and the probabilistic model-based system.

Root Causes	1	2	3	4	5	6	7	8	9
1 No Default	7210		183	39			17		
2 Faulty ONT		3							
3 Configuration Mistake			0						
4 Attenuate drop fiber				72			18		
5 Broken drop fiber					1463				
6 Power Supply shutdown	2					780			
7 Attenuate feeder fiber							0		
8 Broken feeder fiber		1						57	
9 Unknown root cause	716	4		19		27			0

In Table 4, bold numbers placed on the matrix diagonal represent cases for which the expert system and the model have given the same diagnosis result for ONT_1 consistent with the results supervised by the oracle (GPON-FTTH network engineers). For example, as expected by oracle analysis of observations, we note 7210 cases, 3 cases, 72 cases, where the two methods have respectively concluded that no default is identified on the PON, the ONT under test is faulty and the drop fiber of this ONT is experiencing attenuation. See Table 4. We focus our analysis on cases placed outside the diagonal of the matrix, i.e, the cases for which diagnosis results of the two methods do not converge.

There are 183 cases for which the expert system is unable to detect a configuration mistake during ONT provisioning. The reason for this is simply that there is not yet expert rule that handles this situation. Table 5 shows the diagnosis of the probabilistic model for this simple situation.

Table 5: The PON has forty $ONTs$. The $NewONT[1]$ alarm is observed on ONT_1 . Counters and scalar parameters of all $ONTs$ are nominal.

Root causes	States	Beliefs
$ONTConfiguration$	$[OK, \neg OK]$	$[0.020, \mathbf{0.980}]$

We note 39 cases for which the expert system returns no default while the probabilistic model diagnoses that the drop fiber of ONT_1 under test is experiencing attenuation. In order to explain this difference, we looked very closely at the observations collected for these cases. Either the upstream

Table 6: The PON has forty *ONTs*. No alarm is observed. The upstream received power of *ONT*₁ is non-nominal, i.e. node *Rx_{OLT}*[1] has low value. The downstream received power of *ONT*₁ is not measured, i.e, node *Rx_{ONT}*[1] is missing. Counters and scalar parameters of neighboring *ONTs* are nominal.

Root causes	States	Beliefs
<i>Fiber_{DB₁}</i>	[<i>OK</i> , AT , <i>BR</i>]	[1.e-03, 0.93 , 5.e-02]
<i>Fiber_{DB_{i≠1}}</i>	[OK , <i>AT</i> , <i>BR</i>]	[0.92 , 8.e-02, 2.e-06]
<i>Fiber_T</i>	[OK , <i>AT</i> , <i>BR</i>]	[0.99 , 2.e-09, 1.e-48]

received power or the downstream received power of *ONT*₁ is low, and other is missing while the power levels of its neighbors are nominal. The expert rules do not clearly describe this situation, while the model does so naturally. Indeed, expert rule needs both upstream and downstream received powers to diagnose the state of the optical link, while probabilistic model is capable to deals with missing of one received power level since it make a global analysis of the PON. An instance of this case is described by Table 6. Note in Table 6 that the feeder fiber noted *Fiber_T* does not attenuate since power levels of neighbors of *ONT*₁ are nominal. Note that when power levels of neighbors are also low, the probabilistic model understands that it is the feeder fiber of the PON shared by all *ONTs* which is experiencing attenuation (see Table 7). This last situation is encountered in 17 cases for which the expert system does not detect any fault. This situation also explains the 18 cases for which the expert system returns that the drop fiber is attenuating (when both *Rx_{OLT}*[1] and *Rx_{ONT}*[1] are lows) while the model says us that it is rather the feeder fiber which is experiencing attenuation.

Table 7: The PON has forty *ONTs*. No alarm is observed. The upstream received power of *ONT*₁ is non-nominal, i.e. node *Rx_{OLT}*[1] has low value. The downstream received power of *ONT*₁ is not measured, i.e, node *Rx_{ONT}*[1] is missing. Received power levels of neighboring *ONTs* are non-nominal.

Root causes	States	Beliefs
<i>Fiber_{DB_i}</i>	[OK , <i>AT</i> , <i>BR</i>]	[0.62 , 0.37, 2.e-06]
<i>Fiber_T</i>	[<i>OK</i> , AT , <i>BR</i>]	[2.e-02, 0.97 , 9.e-06]

There are 780 cases for which both the expert system and the probabilistic model diagnose that the power supply of *ONT*₁ is down or faulty. See Table 8. Table 9 shows that the most probable values of missing counters

and scalar parameters of the *ONT* in question, computed by the probabilistic model, enforce the belief that the power supply of this *ONT* is shut down or defective. The most probable state of the bias current of this *ONT* is I_0 , i.e., a null bias current. The voltage is null with a belief of 0.95. The transmitted and received powers of this *ONT* have very low values, with a belief of 0.99.

Table 8: The PON has twenty *ONTs*. *DG* alarm is observed on ONT_1 , i.e., node $DG[1]$ takes value 1. Counters and scalar parameters of ONT_1 are missing. Counters and scalar parameters of neighboring *ONTs* are nominal.

Root causes	States	Beliefs
Alt_{ONT_1}	$[OK, \neg OK]$	$[0.001, \mathbf{0.999}]$
$Alt_{ONT_{i \neq 1}}$	$[OK, \neg OK]$	$[\mathbf{0.999}, 0.001]$

Table 9: The most probable states of missing observations useful to detecting shutdown or defective power supply in table 8.

Root causes	States	Beliefs
I_{ONT_1}	$[\mathbf{I_0}, I_1]$	$[\mathbf{0.95}, 0.05]$
V_{ONT_1}	$[\mathbf{V_0}, V_1]$	$[\mathbf{0.98}, 0.02]$
Rx_{OLT_1}	$[\mathbf{Rx_0}, Rx_l, Rx_n]$	$[\mathbf{0.99}, 2.e-03, 1.e-03]$
Rx_{ONT_1}	$[\mathbf{Rx_0}, Rx_l, Rx_n]$	$[\mathbf{0.99}, 4.e-03, 4.e-03]$
Tx_{ONT_1}	$[\mathbf{Tx_0}, Tx_l, Tx_n]$	$[\mathbf{0.99}, 3.e-02, 3.e-03]$

Confusion matrix depicted by Table 4 also shows 2 cases for which the model returns no default, while expert system said that power supply is down. This difference is due to the fact that the expert rule concludes that the power is down when the $DG[1]$ alarm is observed. This narrow rule does not consider the situation where the $DG[1]$ alarm is observed but the counters and scalars of ONT_1 are still available and nominal. The model clearly understands this situation and deduces that power supply of ONT_1 is not down but it is going to be shut down, i.e., there is no default at this time. See Table 10.

The most significant cases for which the probabilistic model clearly outperforms the rule-based expert system are the cases for which the expert system has trouble computing the diagnosis due to missing observations or unforeseen situations, i.e., the situations not yet covered by the existing rules. For each of these cases the expert system returns *unknown root cause*,

Table 10: The PON has twenty *ONTs*. *DG*[1] alarm is observed on *ONT*₁. Counters and scalar parameters of *ONT*₁ are nominal. Counters and scalar parameters of neighboring *ONTs* are nominal.

Root causes	States	Beliefs
Alt_{ONT_1}	[OK , $\neg OK$]	[0.910 , 0.090]
$Alt_{ONT_{i \neq 1}}$	[OK , $\neg OK$]	[0.999 , 0.001]

i.e., it is not able to output any conclusion, while the probabilistic model naturally computes the most probable diagnosis based on available observations even if some of them are missing.

There are $716+4+19+27$ cases for which the expert system fails because observations are not sufficient to run its rules, while the model uses the few available observations in order to complete the observations by inferring the most probable states of useful missing observations. The diagnosis is then computed based on completed observations. For example, there are 19 cases for which the expert system fails to compute the diagnosis based on the observation of the *LEVELLO*[1] alarm with missing received power levels of *ONT*₁. For each of these cases, the probabilistic model infers that the most probable values of these received powers are very low, i.e, Rx_0 in Table 12, and deduces that the most probable explanation of the *LEVELLO*[1] alarm is the drop fiber attenuation of *ONT*₁. See Table 11. We also note among these cases, 716 cases for which the expert system fails while the probabilistic model returns no default. This result may seem trivial, however it has a capital interest for a telecommunication operator. Indeed, this result indicates a nominal operation of the GPON-FTTH infrastructure and allows to deduce that the fault is certainly in the home network of the complaining subscriber. Therefore, this result allows operator to avoid triggering useless interventions on GPON-FTTH infrastructure, but rather to intervene eventually on the home network of the subscriber. So, this result allows to reduce the number of technical interventions and save time and money to customer service of the operator.

Self-diagnostic results of the probabilistic model are more consistent with the behaviour of the GPON-FTTH access network, and more reasonable on a representative sample of cases, than the rule-based expert system. Note for example that for 7% of cases, the expert system fails, while the probabilistic model correctly computes a diagnosis. The principal explanation

Table 11: The PON has forty *ONTs*. The *LEVELLO* alarm is observed on *ONT*₁, i.e., $N_2[1] = LLO$. Upstream and downstream received power of *ONT*₁ are missing. Counters and scalar parameters of neighboring *ONTs* are nominal.

Root causes	States	Beliefs
$Fiber_{DB_1}$	$[OK, \mathbf{AT}, BR]$	$[5.e-02, \mathbf{0.94}, 1.e-06]$
$Fiber_{DB_{i \neq 1}}$	$[\mathbf{OK}, AT, BR]$	$[\mathbf{0.91}, 8.e-02, 1.e-06]$
$Fiber_T$	$[OK, \mathbf{AT}, BR]$	$[\mathbf{0.99}, 2.e-021, 5.e-105]$

Table 12: The most probable levels of missing received powers useful in detecting feeder fiber attenuation in table 11.

Nodes	States	Beliefs
Rx_{OLT_1}	$[\mathbf{R}\mathbf{x}_0, Rx_l, Rx_n]$	$[\mathbf{0.88}, 0.10, 0.02]$
Rx_{ONT_1}	$[\mathbf{R}\mathbf{x}_0, Rx_l, Rx_n]$	$[\mathbf{0.82}, 0.17, 0.01]$

of these differences is that the probabilistic model has a wide scope of reasoning that allows it to make global or generalized intelligent analyses of situations. Doing so, the model covers unforeseen situations encountered by the rule-based expert system, which reasons case-by-case. Therefore the rule-based expert system lacks generalized reasoning capabilities.

Note that in tables above, the numbered root causes correspond to the final diagnosis decision. It is a podium of the most probable root causes that explains observations. This final diagnosis decision is computed from beliefs or posterior probabilities of all root causes of the GPON-FTTH model depicted in Figure 7. Beliefs of root causes are computed by the well-known algorithm of evidence propagation on the junction tree. To compute the final diagnostic decision from root cause beliefs, we proceed in two steps. We start the algorithm by finding the set R_0 of root causes whose the beliefs are not equally probable. This step aims to discard hypotheses for which the model has trouble selecting the most probable state. In the next step, from R_0 , we find the set R of negative hypotheses, i.e., the set of root causes for which the most probable states are negative states. The order of the set R gives us a podium of the most probable root causes identified by the probabilistic model.

In summary we have shown that the Bayesian network model designed in this paper, even with parameters manually and roughly determined by

skilled humans on GPON-FTTH network, gives very satisfying self-diagnosis results of an operating GPON-FTTH network. Nevertheless, we think that we can improve these self-diagnosis results if the parameters, i.e. the conditional probability distributions of the GPON-FTTH network model are fine-tuned by a machine learning algorithm from tremendous amount of data generated by the components of this network. The next section studies EM (Expectation Maximization [30] [32], an algorithm for parameters estimation of a statistical model based on maximum likelihood of the learning dataset.

6 Maximum Likelihood Estimation from incomplete dataset with EM algorithm

Maximum Likelihood Estimation (MLE) can be apply to perform parameters estimation of a statistical model based on a complete or incomplete dataset. Our interest is focused on the case for which dataset is incomplete since real dataset generated by an operating telecommunication network is almost always incomplete. So we focus the study of MLE on EM algorithm [30] [31] which is capable to deals with missing data. However basic concepts on MLE from complete data is detailed in [33] with a particular study of the case of bayesian network model.

6.1 Incomplete dataset

In statistics, missing data occur when no data value is stored for some variables in an observation. This can occur because measurements are not performed properly or because some variables are not reported. In BN some nodes are observations/measurements whereas other nodes are hypotheses or latent variables. Latent variables (as well as hypotheses) are not directly observed but rather inferred from measurements. BN is consequently a setting in which incomplete data occur.

MLE from incomplete data is not straightforward. Indeed most of the time it is not possible to compute the value of the likelihood of the dataset with incomplete data. Indeed let us assume that X is the vector of observed data (or measurements) and Y is a vector of missing data (or latent variables). Computing the joint likelihood $\mathbb{P}(X, Y)$ of the complete data (X, Y) is supposed to be straightforward under the considered model. As Y is not measured it is unfortunately not possible to tune the parameters

of the model by maximizing $\mathbb{P}(X, Y; \theta)$ with respect to θ . Rather, the likelihood of observed data $\mathbb{P}(X; \theta)$ should be maximized with respect to θ . But the computation of $\mathbb{P}(X; \theta)$ is most of the time not tractable. Indeed $\mathbb{P}(X; \theta) = \sum_Y \mathbb{P}(X, Y; \theta)$ and the number of terms in the sum \sum_Y is huge since this is the product of the number of states of each component of the vector Y . The complexity grows exponentially fast with the number of components in Y (e.g. number of nodes that represent a latent variable in the BN). As the likelihood of the observed data $\mathbb{P}(X; \theta)$ is not computationally tractable it is even more an issue to maximize $\mathbb{P}(X; \theta)$ with respect to θ .

6.2 The Expectation Maximization (EM) algorithm

The problem of MLE from incomplete data can be solved with the EM algorithm [30]. As explained above computing the log-likelihood $\log \mathbb{P}(X; \theta)$ of the observed data is not possible, whereas computing the log-likelihood of the complete data $\log \mathbb{P}(X, Y; \theta)$ would be possible if only Y was not missing. It would then be possible to maximize $\log \mathbb{P}(X, Y; \theta)$ with respect to θ . As Y is missing, rather than maximizing $\log \mathbb{P}(X, Y; \theta)$ with respect to θ , the EM algorithm attempts to maximize iteratively the expected value of the log-likelihood of the complete data. Let us introduce $Q(\theta, \theta')$ as follows:

$$Q(\theta, \theta') = \mathbb{E}(L(X, Y; \theta) | X; \theta') \quad (14)$$

In the equation above $\mathbb{E}(\bullet | X; \theta')$ stands for the expected value under the probability distribution of the missing data Y conditionally to the measurements X (for the value θ' of the model parameters set). EM is an iterative algorithm. At each iteration $Q(\theta, \theta^r) = \mathbb{E}(L(X, Y; \theta) | X; \theta^r)$ is maximized with respect to the first parameter θ , that is to say:

$$\theta^{r+1} = \underset{\theta}{\text{Arg max}} Q(\theta, \theta^r) \quad (15)$$

Each iteration is decomposed into two steps: the expectation step (E step), and the maximization step (M step). The E step computes the probability distribution of missing data Y conditionally to the measurements X under the model with parameter θ^r (i.e. the current value of the parameter estimate). In practice this comes down to computing some statistics that summarize this conditional probability distribution, as it is explained in the particular case of the Bayesian network model in [33]. The E step is the most demanding step in the EM algorithm.

In the M step the function $Q(\theta, \theta^r)$ is maximized with respect to the first parameter θ so that an updated value θ^{r+1} of the parameter estimate is obtained. Very often there exists a closed-form solution of this maximization problem so that the problem is simple to solve.

As stated before the EM algorithm is an iterative algorithm. It must be initialized with a value θ^0 . At each iteration the parameter estimate is updated as follows: at iteration 1, $\theta^1 = \text{Arg max}_{\theta} Q(\theta, \theta^0)$, then at iteration 2 $\theta^2 = \text{Arg max}_{\theta} Q(\theta, \theta^1)$ and so on until the algorithm converges to a stable value of θ . It has been proven that each iteration of EM increases the log-likelihood of measurement data, that is to say:

$$L(x; \theta^{r+1}) \geq L(x; \theta^r). \quad (16)$$

As a consequence θ^r converges to a maximum of the log-likelihood $L(X; \theta)$ of measurement data (or a saddle point). It is important to note that this maximum can be a local (but not necessarily global) maximum. In practice this means that the initial value θ^0 must be selected with care in order to avoid problems of convergence to local (but not global) maximum.

The EM algorithm has been adapted to the particular case of bayesian networks [31]. We have performed a detailed study of EM algorithm for BN model in [33]. So, we focus in this paper to show how EM can be apply to fine-tune the parameters roughly determined of our bayesian network model of the GPON-FTTH network for self-diagnosis purposes.

7 Application of EM Algorithm to GPON-FTTH access network

We recall that the reason for which we use EM is that, dataset generated by the components of GPON-FTTH network is not complete, i.e, there are some missing facts. Moreover, we strongly think that, these data, even incomplete, contain relevant knowledge about fault propagation in GPON-FTTH network, that even a proven skilled human on this network does not guess.

7.1 GPON-FTTH network data

The GPON-FTTH network data contain alarms, transmitted and received power of network components, transmission error counters, currents, voltages, temperatures and so on. The dataset corresponds to two months of

measurement on a commercial PON of the Orange FTTH access provider in july-august 2015. It contains 10611 diagnosis cases. A diagnosis case is a combination of value of states or values taken by some variables of the GPON-FTTH model depicted by Figure 7. In practice, there is always some situations where the network management system fails to get some values from some network components. These situations may be due for example to the filtering policy of network data applied by the network operator, or due to the communication loss between the network management system and one or many network components or due to some older devices that may not generate some facts. These situations lead to missing variables. That is why we have used the EM algorithm in order to automatically fine tune the expert parameters of the GPON-FTTH network model. We have divided the dataset into two subsets. The first subset corresponds to 5121 real diagnosis cases collected by Orange on a commercial GPON-FTTH network in july 2015 and the second subset corresponds to 5490 real diagnosis cases collected in august 2015. The first subset is the training dataset used to learn the parameters of the BN model with the EM algorithm. The second subset is the test dataset that we use to assess the performance of fault localization with the fine tuned BN model.

As explained in section 6, the EM algorithm is initialized with a value θ^0 of the parameters vector. In our case θ^0 has been determined from operational expertise in diagnosing GPON-FTTH networks, i.e, θ^0 is the value of parameters vector roughly determined in Section 5.

7.2 Results of the application of EM algorithm for parameters learning of the GPON-FTTH network model

In this section, we assess the benefits of fine tuning the parameters by the EM, with respect to a diagnosis based on a BN which parameters have been set by an expert. For doing so we compare the diagnosis results of PANDA over 5490 experimental cases in two situations: the model parameters have been set by an expert, or they have been moreover fine tuned with an EM by mining the training dataset.

Figure 8 displays the evolution along the first iterations of the EM algorithm of the log-likelihood value of the 5121 real diagnosis cases collected by Orange in july 2015 on an operational GPON-FTTH network. As stated by Equation 16 one can observe on Figure 8 that the log-likelihood of measurement data increases at each iteration of the EM algorithm. The log-likelihood stabilizes at a maximum value after 7 or 8 iterations, and the

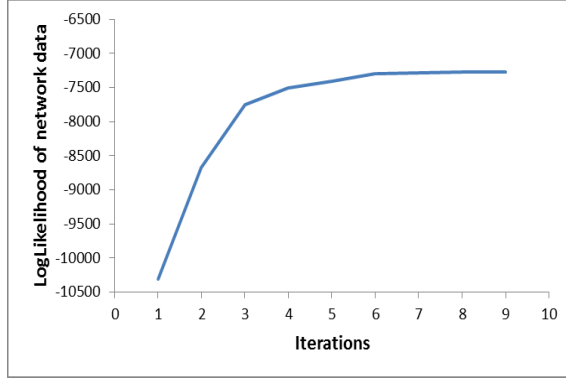


Figure 8: The Evolution of the Loglikelihood of network data with iterations

algorithm then converges. From a practical point of view it is important to mention that we have encountered a few numerical problems with real measurement data. These problems appear when data do not perfectly fit to the theoretical model (which bounds to happen when one deals with real data), for example if the dataset contains outliers [33].

7.3 Diagnosis results of the probabilistic model with optimized parameters by EM algorithm

At that point we had two models: the expert probabilistic model with parameters roughly determined by a skilled human on GPON-FTTH network, and the model with parameters learned by EM, so-called learned model. We have performed the diagnosis on the same cases separately with the two models. Table 13 shows a 2-dimensions confusion matrix which compares self-diagnosis results of the two models. The rows and columns of this matrix respectively represent self-diagnosis carried out with the expert model and with the learned model. Fine tuning the parameters of the model changed the diagnosis results in 185 cases out of the 5490 cases. This means that the order of magnitude of the parameters had been evaluated correctly by the expert. However it is interesting to analyze into more details cases where the diagnosis was different with a finely tuned BN, i.e. the cases corresponding to non-zero values outside the diagonal of the confusion matrix. For example Table from 14 and 15 compare the two models for two instances of those cases. The title of each of these tables is a short description of observations collected on the operating GPON-FTTH network for the considered case. Note that the comparison of the two models for each instance of those cases

is done in [33].

Table 13: 2-Dimensions Confusion Matrix of self-diagnosis results between the two models.

Root causes	1	2	3	4	5	6	7	8	9
1 No Default	4030			7	6		9		
2 Configuration Mistake		183							
3 Faulty ONT			0						
4 ONT power supply				402					
5 Drop fiber attenuated					56				
6 Drop fiber broken					14	602	1		
7 Feeder fiber attenuated			148				32		
8 Feeder fiber broken								0	
9 Unknown root cause									0

Table 14 shows that, on a *PON* of forty *ONTs*, when the upstream received power of an *ONT* denoted by ONT_1 is low while the downstream received power of ONT_1 is nominal, the expert model does not detect any fault. This is a wrong diagnosis carried out by the expert model since the upstream optical channel between the *OLT* and ONT_1 is experiencing attenuation. On the other hand, the learned model computes the appropriate diagnosis, i.e. attenuation of the drop fiber $Fiber_{DB_1}$ of ONT_1 . Note that *OK*, *AT* and *BR* denote a fiber which does not attenuate, which attenuates or a broken fiber. This situation appears in 6 cases in the test dataset.

Table 14: *PON* with forty *ONTs*. No alarm on the *PON*. Upstream received power $RxOLT[1]$ of ONT_1 is low. Downstream received power $RxONT[1]$ of ONT_1 is nominal. Received powers of neighbor *ONTs* are nominal.

Model	Root causes	States	Beliefs
Expert	$Fiber_{DB_1}$	[OK , <i>AT</i> , <i>BR</i>]	[0.9 , 8.e-02, 3.e-06]
	$Fiber_{DB_{i \neq 1}}$	[OK , <i>AT</i> , <i>BR</i>]	[0.9 , 8.e-02, 3.e-06]
Learned	$Fiber_{DB_1}$	[<i>OK</i> , AT , <i>BR</i>]	[9.e-02, 0.9 , 2.e-06]
	$Fiber_{DB_{i \neq 1}}$	[OK , <i>AT</i> , <i>BR</i>]	[0.9 , 8.e-02, 3.e-06]

Table 15 shows a case for which the received power levels of ONT_1 are nominal while those of neighbors of ONT_1 are low. In this situation the expert model diagnoses that the drop fiber of each neighbor of ONT_1 experiences attenuation. Doing so, the expert model assumes that when the received power levels of at least one *ONT* on the *PON* are nominal, then

the feeder fiber (denoted by $Fiber_T$) shared by all ONT s connected on the PON cannot experience attenuation although the received power levels of other ONT s are low. This reasoning is not always true since for this diagnosis case it is the feeder fiber that attenuates. But this attenuation has not affected ONT_1 since its received power levels were very high before the beginning of the feeder fiber attenuation. On the other hand, the received power levels of neighbors of ONT_1 were nominal but very close to the lower bound of the range of nominal power values. We have observed 9 occurrences of this case.

Table 15: The PON has forty ONT s. No alarm is observed on the PON. The upstream and downstream received powers $RxOLT[1]$ and $RxONT[1]$ of ONT_1 are nominal. The received powers of neighbors of ONT_1 , i.e., $RxOLT[i]$ and $RxONT[i]$ for $i \in \{2, \dots, 40\}$ are low.

Model	Root causes	States	Beliefs
Expert	$Fiber_{DB_1}$	[OK , AT , BR]	[0.9 , 8.e-02, 3.e-06]
	$Fiber_{DB_{i \neq 1}}$	[OK , AT , BR]	[8.e-02, 0.9 , 3.e-06]
Learned	$Fiber_{DB_1}$	[OK , AT , BR]	[0.9 , 9.e-02, 2.e-06]
	$Fiber_T$	[OK , AT , BR]	[5.e-03, 0.99 , 5.e-39]
	$Fiber_{DB_{i \neq 1}}$	[OK , AT , BR]	[0.68 , 0.31, 0.001]

Tables 13, 14 and 15 show that the learned model reasonably improves self-diagnosis on GPON-FTTH network, previously carried out by the expert probabilistic model, and Table 4 shows that expert probabilistic model outperforms the rule-based expert system.

8 Conclusion

We have presented in this paper a comparative study between rule-based expert systems and probabilistic model-based approaches for self-diagnosis of telecommunication networks. The study shows why and how probabilistic reasoning improves self-diagnosis of telecommunication networks compared to rule-based expert systems. We have presented self-diagnosis results obtained with our designed probabilistic model of the GPON-FTTH access network, as well as the results of the comparative study to this network. A 2-dimension confusion matrix shows that the probabilistic model-based method outperforms the rule-based expert system for carrying out self-diagnosis of a large-scale operating GPON-FTTH access network. Nevertheless, we noticed that the probabilistic model-based method can be used

to complete and improve rules of the expert system. But why make the choice to do so? Indeed, a probabilistic GPON-FTTH network model already embeds powerful computational capabilities useful in performing efficient self-diagnosis on this network.

With the main goal to improve performances of the GPON-FTTH probabilistic model, we have studied and applied EM (Expectation Maximization) algorithm in order to automatically fine-tune parameters of this model manually and approximatively determined from GPON-FTTH network skills. As expected, the new probabilistic model with optimized parameters improves self-diagnosis comparatively to the previous probabilistic model. Note that the use of EM in this way may be very interesting for telecommunication operators. For example, a practical application of this work could be to implement a fully autonomous self-diagnostic system in which a closed loop is created between the probabilistic self-diagnosis model integrating machine learning capabilities, diagnostic results leading to interventions on network infrastructure and data reported by the technicians of interventions that will be used to automatically adjust the probabilistic model to make it more efficient. However, the reliability of these data must be guaranteed to avoid doing rather derive the model instead of improving it.

An interesting perspective of this work would be to automatically fully adjust the entire expert probabilistic model (dependencies and parameters) with the aim to further improve diagnostic results. An algorithm such as SEM (Structural EM) could be applied in this case. We also plan to refine and upgrade the probabilistic model in order to consider GPON network evolutions like xGPON, NG-PON (Next Generation PON) and NG-PON-2.

References

- [1] M Steinder and A S Sethi. A survey of fault localization techniques in computer networks. *Science of Computer Programming*, 53:165–194, January 2004.
- [2] Carol Hounkonnou. *Active self-diagnosis in telecommunication networks*. PhD Thesis, European University of Brittany, University of Rennes 1, INRIA, ISTIC, France, 2013.
- [3] L F Pau. Survey of expert systems for fault detection, test generation and maintenance. *Expert Systems*, 3:100–110, April 1986.

- [4] W T Scherer and C C White. *Knowledge-Based System Diagnosis, Supervision, and Control. Chapter 16: A survey of expert systems for equipment maintenance and diagnostics*. Springer US, 1989.
- [5] R D Gardner and D A Harle. Alarm correlation and network fault resolution using the Kohonen self-organising map. *Global Telecommunications Conference (GLOBECOM 1997)*, pages 1398–1402, 1997.
- [6] R J Patton, J Chen, and T M Siew. Fault diagnosis in nonlinear dynamic systems via neural networks. *International Conference on Control*, 2:1346–1351, 1994.
- [7] A Goel, J Ramanujam, and P Sadayappan. Towards a ‘neural’ architecture for abductive reasoning. *IEEE International Conference on Neural Networks*, pages 681–688, 1998.
- [8] Lundy Lewis. A case-based reasoning approach to the resolution of faults in communication networks. In *Proceedings of the third international symposium on Integrated network management*, pages 671–682, 1993.
- [9] Serge Romaric Tembo, Jean Luc Courant, and Sandrine Vaton. A 3-layered self-reconfigurable generic model for self-diagnosis of telecommunication networks. *IEEE SAI International Conference on Intelligent Systems, INTELLISYS, London*, 2015.
- [10] Judéa Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
- [11] Patrick Naim, P H Willemin, Philippe Leray, O Pourret, and A Becker. *Réseaux Bayésiens*. EYROLLES, 2008.
- [12] Antoine Cornuéjols and Laurent Miclet. *Apprentissage Artificiel, Concepts et Algorithmes*. EYROLLES, 2013.
- [13] Telecommunication Standardization Sector of ITU (International Telecommunication Union). *G.984.3 Recommendation*. ITU-T, 2008.
- [14] Telecommunication Standardization Sector of ITU (International Telecommunication Union). *G.977.1 Recommendation*. ITU-T, 2003.
- [15] B Gruschke. Integrated event management: Event correlation using dependency graphs. *A.S. Sethi (Ed.), Ninth International Workshop on Distributed Systems: Operations and Management, University of Delaware, Newark, DE*, 87:130–141, October 1998.

- [16] S Kätker. A modeling framework for integrated distributed systems fault management. *C. Popien (Ed.), Proc. IFIP/IEEE International Conference on Distributed Platforms, Dresden, Germany*, pages 187–198, 1995.
- [17] K Houck, S Calo, and A Finkel. Towards a practical alarm correlation system. In *Proceedings of the fourth international symposium on Integrated network management*, pages 226–237, 1995.
- [18] J F Jordaan and M E Paterokl. Event correlation in heterogeneous networks using the osi management framework. In *Proceedings of the third international symposium on Integrated network management*, pages 683–695, 1993.
- [19] S Kätker and K Geihs. A generic model for fault isolation in integrated management systems. *Journal of Network and Systems Management*, 5(2):109–130, 1997.
- [20] S Kätker and M Paterok. Fault isolation and event correlation for integrated fault management. In *Proceedings of the fifth international symposium on Integrated network management*, pages 583–596, 1997.
- [21] G Jakobson and M Weissman. Real-time telecommunication network management: extending event correlation with temporal constraints. In *Proceedings of the fourth international symposium on Integrated network management*, pages 290–301, 1995.
- [22] José Manuel Sanchez, Imen Grida Ben Yahia, and Noel Crespi. Self modeling based diagnosis of software defined networks. *1st Conference on network softwarization (NetSoft)*, 2015.
- [23] Guanglei Liu and Chuanyi Ji. Resilience of all-optical network architectures under in-band crosstalk attacks: A probabilistic graphical model approach. *IEEE Journal on selected areas in communications*, 25(4), 2007.
- [24] Armen Aghasaryan, Eric Fabre, and Albert Benvenist. Fault detection and diagnosis in distributed systems: An approach by partially stochastic petri nets. *Journal of Discrete Event Dynamic Systems: Theory and Applications*, 8:203–231, 1998.
- [25] Chava Vijaya Saradhi and Suresh Subramaniam. Physical layer impairment aware routing (pliar) in wdm optical networks: issues and

- challenges. *IEEE Communications Surveys & Tutorials*, 11(4):109–130, 2009.
- [26] Stephen Lauritzen. Graphical models. *Oxford Statistical Science Series, Book 17*, Clarendon Press, Oxford, 1996.
 - [27] A L Madsen and F V Jensen. Lazy propagation: a junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113:203–245, 1999.
 - [28] Stephen Lauritzen and David Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.
 - [29] A Berry, P Heggernes, and Geneviève Simonet. The minimum degree heuristic and the minimal triangulation process. *H. Bodlaender. WG’03: 29th International Workshop on Graph Theoretic Concepts in Computer Science, Elspeet (The Netherlands), Lecture Notes in Computer Science lirmm-00191916*, pages 58–70, 2003.
 - [30] A P Dempster, N M Laird, and D B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
 - [31] Stephen Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.
 - [32] Stephen Lauritzen and David Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.
 - [33] Serge Romaric Tembo, Sandrine Vaton, Jean Luc Courant, and Stephane Gosselin. A tutorial on the em algorithm for bayesian networks: application to self-diagnosis of gpon-ftth networks. *IEEE Technically Sponsored Workshop TRAC 2016, Traffic analysis and Characterization*, 2016.

Authors biography



Serge Romaric Tembo is a PhD student at Orange Labs (Lannion, France) and Telecom Bretagne (Brest, France). He is a member of the IRISA laboratory. His research focuses on artificial intelligence and probabilistic reasoning applied to the autonomic and cognitive management of telecommunication networks.



Sandrine Vaton is professor at Telecom Bretagne (Brest, France) and a member of the IRISA laboratory. She holds a PhD in signal processing from Telecom Paristech and an accreditation to supervise research in computer science from the University of Rennes 1. Her research interest involve network monitoring, traffic characterization and performance evaluation among others. She has supervised about 10 PhDs. She has been involved in many research projects at European or French national level, with South America or in bilateral collaboration with the industry. She teaches cryptography and network security, stochastic simulation methods, and queuing systems. She is gender equality adviser at Telecom Bretagne.



Jean-Luc Courant received PhD in microelectronics from the University of Paris VII in 1990. Since 2001, he has been in charge at France Telecom R & D of studies on optical metrology, and since 2007 new diagnostic tools for networks at Orange labs Networks.



Stéphane Gosselin joined the R & D of France Telecom in 1993. He is currently in charge of a research project on network operations and management in Orange Labs in Lannion, France, and is the technical leader of European project COMBO. He authored or co-authored about 70 papers or communications.



Michel Beuvelot is an MBA and Engineer graduate. His career started in France Telecom, pursued in Corvis, a telecommunication systems supplier and later in the Open Group where he led applications development teams. He now serves again Orange in its Innovation Marketing and Technologies division, as an R & D team manager.