ソフトウェア工学　121－5
(1998. 11. 5)

WANに対する確率的な故障診断と修復のための知的エージェント

ハサン ハジ　B.H. ファー　河野善彌

〒338-8570 埼玉県浦和市下大久保255
埼玉大学情報システム工学科
Tel. 048-858-3489,　Fax. 048-858-3716
E-mail. hajji@cit.ics.saitama-u.ac.jp
http://www.cit.ics.saitama-u.ac.jp/~hajji/

あらまし:　現在使われているほとんどのネットワークは、様々なプロトコルモジュール群の組合わせにより実現されている。ネットワークは、必要性の増大に伴い、より複雑になりサイズも巨大化している。よって故障による影響は深刻となり、いかにに故障を出さないかが重要視されている。本研究では、分散化されたベイズネットワークを用いた自動的な故障診断と修復システムを提案する。WAN内の各領域に対して一つのエージェントを割当てる。各エージェントは、固有のの知識をBayesianサブネットとしてモデル化し、他のエージェントと通信することにより故障と修復を行う。各エージェントが管理しているオブジェクトに対して異常が通知された時点で、その異常通知を考慮して最適な修復プランを計算する。次に、未観察のオブジェクトの中の近視眼的な情報の価値(myopic value of information)を用いて逐次的に評価し、最も修復プランのコストを減らす観察オブジェクトを見つけ出す。各エージェント間に「インターフェースセット」というオブジェクトの集合を作成し、異なった領域に存在する各エージェントに関連した情報を見失わない役割を果たしている。これらのインターフェースセットを用いてエージェント間の通信を実現する。

キーワード　故障診断, *WAN*, ベイズネットワーク, 意思決定理論的な故障診断と修復プラニング

Intelligent Agents for Probabilistic Diagnosis-Restoration Planning in WAN

Hassan Hajji　B.H. Far　Zenya Koono

Department of Information and Computer Science, Saitama University
255 Shimo-okubo, Urawa 338-8570, Saitama, Japan
Tel. +81-48-858-3489,　Fax. +81-48-858-3716
E-mail. hajji@cit.ics.saitama-u.ac.jp
http://www.cit.ics.saitama-u.ac.jp/~hajji/

Abstract: As networks increase in size, heterogeneity and complexity, the need to maintain their availability and reliability grows in both importance and difficulty. In this paper, we propose a probabilistic distributed approach to automatic network fault diagnosis and correction in Wide Area Networks (WANs) using Bayesian networks. Each sub-domain of a WAN is assigned to a single agent that models its own partial knowledge as a Bayesian sub-net. When the agent is notified of an anomaly in one of its managed objects, it computes an initial optimal observation plan given reported abnormal observations. The plan specifies the order in which the objects are to be observed. Then, information gathering objects are sequentially evaluated using the myopic value of information, and the agent decides upon the object to observe based on its potential to reveal useful information about which component is faulty, and associated cost. To derive the globally optimal restoration plan, agents maintain an *interface set* of objects through which they coordinate observation and restoration actions, and keep track of relevant information in other domains.

Key words　*Fault management, wide area networks, Bayesian networks, decision-theoretic diagnosis-restoration planning*

1 Introduction

Wide Area Networks (WANs) represent the backbone of modern enterprises communication infrastructure. Faults and performance inefficiencies in these systems give raise to considerable business losses. Hence, tools and approaches for automating the process of identification, correlation, and correction of faults should evolve to meet the need of current and future communication environments [1, 2, 4].

Network fault management is one of the functional areas defined by the OSI management standards. It concerns itself with the detection, diagnosis and correction of anomalous conditions that occur in the network. It is a complex task, due primarily to the fact that it has to deal with the heterogeneity and distributed aspect of networks. Furthermore, a single fault results in a big number of symptoms making it difficult to determine the primary cause of the observed abnormal behavior. Approaches commonly adopted to network fault management relies on techniques developed in Artificial Intelligence. Current existing systems cover expert systems [5], Petri Nets [6], dependency graph models [7, 15].

In this article, we describe a computational paradigm to decentralize and automate the fault diagnosis and restoration of services in WAN. Our approach is based on recent advances in probabilistic reasoning. We model our fault diagnosis problem using Bayesian Networks (BN) [10]. Uncertainty in communication networks raises from the fact that a network experiencing a fault may be under stress, leading to packet loss and potentially loss of information relevant to identifying the nature of the fault. In our approach, we do not model the whole domain in a single BN, but rather assign each sub-domain to a single agent. The rationale behind this choice is to lift the computational complexity of reasoning with BN. Moreover, WAN are naturally arranged as federated multi-domain organizations, where each sub-domain is managed by local administrators. In our model, each agent holds a partial belief about its managed objects. It has the ability to autonomously sense and correct its own environment. Agents keep track of relevant information in other domains by maintaining an *interface set*, and inter-agent communication permits computing globally the optimal restoration plans.

The rest of this paper is arranged as follows: section 2 states the problem, and briefly reviews BN. Section 3 presents the modules of the proposed distributed diagnosis and restoration framework. We conclude in section 4.

2 Problem definition and Bayesian network model

2.1 Problem Statement

As networks grow larger, the accurate diagnosis of faults gets difficult. A single fault in a WAN results in a big number of symptoms making it difficult to determine the primary cause of the observed abnormal behavior. The network devices in such WANs are monitored by data collectors which produce status reports suggesting which device is faulty. Because the network is used for its own surveillance, and because faults are often transitory, these reports are very uncertain evidences of a fault and its location.

The process of isolating and correcting faults is a sequential process in nature. Effective network fault management should not stop at the level of mapping currently observed alarms to probabilities over the state of possible faults, but rather dynamically change these probabilities as new insight is gained during the fault isolation process. More importantly, the ultimate goal is to restore the services of the network, and not produce estimates of which component is faulty. The Network management application should assist the user to make sense of the computed probabilities and suggest which action is going to contribute considerably to restore the services incurring a minimal expected cost. Finally, it is desirable that our fault management application could scale well to large networks. The problem we attempt to solve is this paper can be formulated as:

Given a model for the network, and a set of observed alarms, how one does design an algorithm that can create a number of fault candidates and assign to each a probability to be the cause of the observed anomaly? The algorithm should suggest which action to take next, and in the light of newly acquired information, change its belief about which component is the root cause *of the observed abnormal behavior. The algorithm should be able to scale up to large WANs.*

The algorithm should handle uncertain and contradictory information. It is desirable that it runs in a reasonable time.

2.2 Bayesian Network model

Data communication networks are hierarchically organized structures. They can be modeled at different levels of abstraction which are dictated by the desired application and/or the available information. Data communication network consist of a number of managed objects that have a separate and distinct existence (e.g. routers, bridges, hubs... etc). Objects in such networks are dependant on each other rather in a complicated way. The knowledge of this dependency is very important for effective fault management. Our approach to model such knowledge uses Bayesian network [12]. BN provides a concise representation of the modeled domain. It consists of two parts:

- *A qualitative part:* a graphical representation of the relationship between the variables in the probability distribution that is being represented. This part is a Directed Acyclic Graph (DAG) where each node represents a random variable that can take its values from a pre-defined set of possible values. The arcs, formally, model the dependencies between variables, but it can be, safely, viewed as depicting the cause-effect relationships between variables. Absence of an arc between two nodes means that the corresponding variables do not directly influence each other, and hence, are independent. The DAG captures the cause-effect relationships between objects in the network.

- *A quantitative part:* a set of Conditional Probability Tables (CPT) associated with each parent-child cluster. The CPTs together provide an economical decomposition of the joint probability. This CPTs quantify the causal strength between each parent and its children.

Figure 1 shows an example of a simplified network problem. Once the Bayesian network is

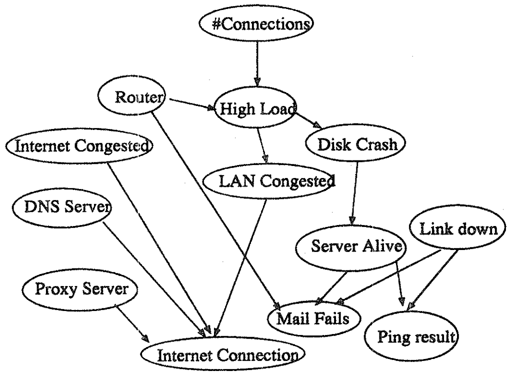

Figure 1: Example of a simple Bayesian network

constructed, it can be used as an inference engine to infer the probabilities that a given object is the cause of the observed abnormal behavior. The appendix summarizes belief update algorithm in singly connected networks due to Pearl [10, 12]. This algorithm is known to have a polynomial runtime. In arbitrary networks, The algorithm most commonly was designed by Lauritzen Spiegelhalter, and later refined by Jensen [14]. The algorithm compiles the initial BN into a second structure called *junction tree* , through a number of graphical operations. The nodes of a junction tree, called *cliques*, are subsets of nodes of the initial BN. After entering an observation in the appropriate clique, a *message passing* between neighboring cliques in the junction tree results in each clique holding the joint distribution of its variables. To calculate the posterior probability of some node of interest, the joint probability is marginalized over other variables sharing the same clique with the node of interest. Note however, that inference in arbitrary Bayesian networks is NP-Hard [13].

3 Distributed probabilistic fault diagnosis and restoration

Effective network management requires distributing processing and control activities among different agents. Under a distributed architecture, the system would benefit a lot in terms of network traffic, overload of the management platform and accuracy of the fault isolation process. Decentralizing control among agents is also mo-

tivated by the need to speed up inference in BN. We first discuss how agents can be assigned to different domains. Then, we look at the problem of restoration planning from a single agent view. We end this section by discussing the issue of restoration planning from the multi-agent perspective.
The distributed structure of our approach is arrived at by assigning each sub-domain to a single agent that models its partial knowledge as a Bayesian sub-net (Figure 2).

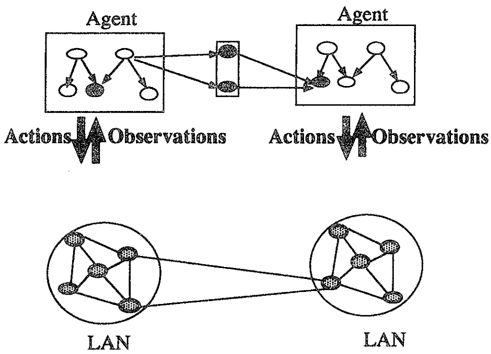

Figure 2: A model for distributing processing and control among multiple agents

Typically, observations made by an agent within its domain might be symptoms of a faulty object in another sub-domain. It is, then, important to make sure that the partitioning of the WAN allows this information to propagate to the concerned agent. To meet this requirement, we notice that the apparent dependency between many objects in different domains might be induced by a small number of objects standing high in the causal chain. If we could construct this set and allow each two agent to keep a copy of it, we are sure that no information is lost. Furthermore, the set needed to keep each two agents from influencing each other's belief is exactly the set of nodes that d-separates [1] their domains. It follows then that by searching the set d-separating each two agent's sub-domains, we guarantee that no information is blocked during inference in BN. We call this set an *interface set* (the black circle in figure 2). A formal proof of this result can be found in [11].

[1]A set of nodes Z d-separates the sets X and Y, if X and Y are conditionally independent given Z

Once the local Bayesian sub-nets are constructed, each agent compiles its own model into a junction tree, and the junction trees are interconnected through the interface sets.
This approach has a number of advantages. WAN are naturally arranged as federated multi-domains according to some geographical or organizational criteria. It make sense, then, to process, filter symptoms first within the domain, and afterwards process the results between domains. For example, in the example of figure 1, the Backbone Router (router, DNS server, Proxy server, physical links ...etc) domain may be managed by an Internet Service Provider, while local administrators would typically manage the LAN domain. Furthermore, an agent containing a faulty object may not be aware of it within its own domain. By cooperatively communicating with other agents, it can benefit from various views of the same problem, leading to more efficient fault restoration schemes.

3.1 Single Agent Restoration Planning

In this section, we present an algorithm for scheduling observation and/or repair actions within the domain of a single agent. We assume that the agent manages n objects that we denote by $\{ O_1, \ldots, O_n \}$. Each object O_i can be exactly in either of two states: it is working properly, denoted by $O_i = ok$, or it is faulty, denoted by $O_i = \neg ok$. Let p_i denote the probability the object O_i is the faulty given the notification. The set of objects an agent can manage is represented as nodes of a Bayesian sub-net. The nodes are divided into two types:

- Fault Nodes: these nodes represent the object that might be a cause to the observed abnormal behavior. Each fault node O_i can be observed incurring a cost c_i^o, and if the object is found faulty, it is repaired incurring a cost c_i^r. The costs may denote an estimation in terms of monetary units of the resources, for example number of human operators, time consumed, and money needed to observe the object O_i.

- Sensor Nodes: These nodes represent objects of the network that do not confirm any system failure, but can be consulted to gain more information about faulty objects. This type of nodes are observable, but not eligible for repair. Each sensor node O_j is observed incurring a cost c_j^o.

The problem facing the agent is to determine a restoration plan that orders the observation and/or repair actions, such that the faulty object is located with minimal expected cost.

If we observe the objects sequentially in the order $\{O_1, \ldots, O_n\}$, that we denote by $\Pi(1, \ldots, n)$ (or simply Π when there is no risk of confusion), depending on which object is faulty, we incur different costs up to locating the faulty object. Specifically, under the plan $\Pi(1, \ldots, n)$, we pay $c_1^o + c_1^r$ if O_1 is the only faulty object; $c_1^o + c_2^o + c_2^r$ if O_2 is the only faulty object in the set $\{O_2, \ldots, O_n\}$. Generally, we incur the cost $c_1^o + c_2^o + \ldots + c_i^o + c_i^r$ if and only if O_i is the only faulty object in the set $\{O_i, \ldots, O_n\}$. This event occurs with the probability $P(C_i = \neg ok, C_{i+1} = ok, \ldots, C_n = ok)$. Hence, the expected cost of restoration under plan $\Pi(1, \ldots, n)$, denoted $EC(\Pi)$, is obtained by weighting the costs that we would possibly incur under plan $\Pi(1, \ldots, n)$ with the probability of that event to happen. Furthermore, if we assume that there is only a single fault, the expression of the expected cost simplifies to:

$$EC(\Pi) = \sum_{i=1}^{n} \Big(\sum_{k=i}^{n} p_k \Big) c_i^o + \sum_{i=1}^{n} p_i c_i^r \qquad (1)$$

Note that $\sum_{i=1}^{n} p_i c_i^r$ does not depend on the ordering. Consequently, we need to minimize only the first part of equation 1. Furthermore, note that for any given plan $\Pi(1, \ldots, i, i+1, \ldots, n)$, swapping the position of any two consecutive objects, say O_i and O_{i+1}, and keeping the position of all other objects intact, we get another restoration plan $\Pi(1, \ldots, i+1, i, \ldots, n)$ that is dominated by $\Pi(1, \ldots, i, i+1, \ldots, n)$ if its expected cost $EC(\Pi(1, \ldots, i, i+1, \ldots, n))$ is bigger than the expected cost of the initial plan $EC(\Pi(1, \ldots, i+1, i, \ldots, n))$. That is to say :

$$\begin{aligned} &EC(\Pi(1, \ldots, i, i+1, \ldots, n)) - \\ &EC(\Pi(1, \ldots, i+1, i, \ldots, n)) \leq 0 \end{aligned} \qquad (2)$$

This expression can be easily simplified to the equation derived by Kalagnanam & Henrion [8]:

$$\frac{p_i}{c_i^o} \geq \frac{p_{i+1}}{c_{i+1}^o} \qquad (3)$$

It follows that, by sorting the objects according to probability-to-observation-cost ratio, we obtain an optimal observation plan. For instance, if the cost of observation are the same, we first observe the object with high probability of failure.

Note that some objects fault nodes may not be observable. Accounting for this case is straightforward: each unobservable fault node is observed with its repair cost and always found faulty.

3.2 Integrating isolation and restoration planning

Up to this point we restricted the action available to the agent to be only observation of objects before eventual repair. In many cases, the agent has the possibility to take sensing action that may reduce its uncertainty about the state of the world. For example, an agent suspecting a link failure may run the *ping* utility, and thus reduce its uncertainty about the objects susceptible to be the root cause of the observed anomaly. In this subsection, we integrate fault diagnosis and the restoration planning in a single process. We show how to map a set of observation in the agent's domain to an optimal plan. The proposed approach is based on an approximate method for evaluating information-gathering actions [9].

Let us assume that the agent has m possible sensors nodes $S_1, \ldots, S_n$ that it can consult to reduce the uncertainty about which object is faulty. Furthermore let $\{s_1^j, \ldots, s_{l_j}^j\}$ be the set of possible values S_j can take, and let $\Pi(\emptyset)$ denote the restoration plan initially computed, assuming no sensing information. Recall from section 3.1, that $\Pi(\emptyset)$ can be derived by ordering the objects according to their probability-to-observation-cost ratio. The problem is to determine at each decision point whether to try an observation action, as dictated by $\Pi(\emptyset)$, or observe the value of some sensor node. The aim is to keep the cost of restoration as cheap as possible.

The merit of querying a sensor node S_j can be decided before actually observing its value, using the following consideration: if we observe S_j and find the value to be s_i^j, the gain from this sensing action is the cost of the restoration plan given that the agent observed S_j and found that its value is s_i^j minus the expected cost of restoration plan when the value about S_j is not known. It is clear that this quantity is positive, since an agent that knows the state of S_j will restore the network cheaper than an agent without this information. However, since we are not sure of the actual outcome of querying S_j, we must average over all possible outcomes of S_j. We write:

$$EC(\Pi(S_j)) = \sum_{k=1}^{l_j} \left(\Pi(s_k^j)P(S_j = s_k^j)\right) \qquad (4)$$

$\Pi(S_j)$ denote the improvement in the restoration plan if the agent decides to query the value of S_j. Let $c^o_{S_j}$ denote the cost of observing S_j. Adding this cost to the improvement $EC(\Pi(S_j))$, may not remain smaller than the expected cost of $\Pi(\emptyset)$. That is to say, the improvement brought by S_j may not justify its cost. To identify the best information gathering action, the above process is iterated for all the sensors, and the agents ends by selecting to observe the sensor with a minimal improvement in the restoration plan even if the cost of observation is added. If such sensor is not found, the object with high probability-to-observation-cost is observed and possibly repaired if found faulty.

The following algorithm summarizes the result of the previous two subsections:

Diagnosis-Restoration

(Observed Symptoms) *Returns* Action

1. identify potential faulty objects not yet observed or repaired.
2. Order the ratios $\frac{p_i}{c_i^o}$ of components not yet observed or repaired , and let $\Pi(\emptyset)$ the resulting order.
3. For all the sensors S_j not yet observed
4. For All $s_i^j \in \{s_1^j, \ldots, s_{l_j}^j\}$
   (a) Submit $S_j = s_i^j$ as observation to the agent Bayesian sub-net
   (b) calculate the optimal restoration plan using equation (4)
   (c) calculate the restoration plan $\Pi(S_j)$, if S_j is to be observed using equation (4).
5. End For
6. Find the plan with minimal expected cost among the quantities $EC(\Pi(S_j)) + c^o_{S_j}$ and $EC(\Pi(\emptyset))$.
7. End For
8. Suppose that O_i is the object with highest observation cost-probability ratio, not yet observed or repaired. If there exist a sensor such that $\Pi(S_j) + c(S_j)$ is minimal, then return OBSERVE-S_j, Otherwise return REPAIR-O_i.

3.3 Multi-agents diagnosis-restoration planning

In the previous section, we showed how observations in the agent's domain are mapped into actions. This section explains the inter-agent communication, and how the contents of messages are used to coordinate agents actions.

As explained earlier, agent influence each other's belief through the interface set. Note that an agent may have in one of its interface sets, objects that it does not manage. To illustrate how agent communication is performed, assume that we have only two agents: A and B, and suppose that agent A has been notified of some abnormal behavior in its managed domain. Agent A first determines the probability of each object to be the root cause of the anomaly, and consequently it updates the probability of all the managed objects including the objects in the interface set. Agent B, notices this change[2], and launches a diagnosis-restoration activity. It calculates the expected cost of restoring its sub-domain and broadcast it to agent A. The sum of this two quantities is indexed by the corresponding action and, then, stored. This process continues during all the time agent A evaluates its sensors. Once, this evaluation is finished, agent A sorts its memory, and returns the plan with minimal cost to agent B. Let us denote the expected cost of this restoration plan by α. Agent B, in turn, begins investigating the possibilities to improve α. For all sensors the agent B can access, it calculates its restoration plan, and asks agent A to send its restoration plan, computed given the observed changes in the interface set. This two quantities are summed, and stored only if the sum is smaller than the quantity α previously sent by A. Once communication ends, agent B takes an action if

[2]Note this change may not be a certain observation. BN allows even changes in the probability distribution to be entered as a special type of observation called *soft evidences*

it succeeded to find a restoration plan with expected cost smaller than α, otherwise agent A takes the appropriate action, namely action indexed by the quantity α.
The process we explained above allows the agents to restore the network with a globally optimal cost. We restricted the discussion to the case of two agents, but generalization to multi-agents is straightforward.

4 Conclusion

In this paper, we presented a distributed probabilistic framework for fault diagnosis and restoration in WAN. The proposed approach differs from earlier network management methods in two major ways: it explicitly recognizes the incompleteness of information inherent in communication networks [3, 7]; it interleaves diagnosis and restoration in a single process : the ultimate goal is to restore the network, and information gathering actions are worthwhile only to the extend they honor this goal. The assumption of single fault is reasonably realistic: more than one objects are unlikely to fail in the same time. Extending to the general of dependent faults can be computationally expensive. The concepts discussed in this paper are being realized in a network management system currently under active development.

APPENDIX

Pearl's algorithm views each node as an individual processor. Each node performs local computation, and the results are communicated only to neighboring nodes. A typical fragment of a singly connected network is shown is Figure 4. The conditional probabilities $P(x|u_1,\ldots,u_n)$

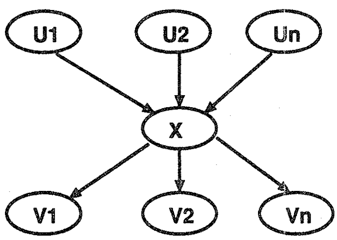

Figure 3: An example of multi-domain WAN

quantitatively relate the node X to its parents. Let $W^-_{XV_j}$ denote the evidences (the observed value of nodes) on the head side of the arc $X \rightarrow V$, and $W^+_{U_iX}$ denote the evidence in the subnetwork on the tail side of the arc $U_i \rightarrow X$. The total evidence is given by $W = \{W_{X^-}, W_{X^+}\}$, where $W_{X^-} = \{W^-_{XV_1},\ldots,W^-_{XV_m}\}$ and $W_{X^+} = \{W^-_{U_1X},\ldots,W^-_{U_mX}\}$. Note that for singly connected networks, all $W^-_{XV_i}$ and $W^+_{U_iX}$ are disjoint. In figure 4, the π message

$$\pi_X(u_i) = P(u_i|W^+_{U_iX}) \tag{5}$$

is the current strength of the causal support contributed by incoming arc $U_i \rightarrow X$, and λ message

$$\lambda_{V_j}(x) = P(W^-_{XV_j}|x) \tag{6}$$

is the current strength of the diagnostic support contributed by each outgoing arc $X \rightarrow V_j$
Belief update algorithm ([10, 12])
A node X is activated when it receives the π messages from its parents, the λ messages from its children, or the node itself is instantiated for a specific value x. Upon the activation, X performs the following three times in any order.
Step 1: Initialization

1. Set all λ messages and π messages to 1,
2. For all roots U, set $\pi(u) = P(u)$.
3. For all roots U and all children X of U, the node U posts new π messages to X:

$$\pi(x) = \begin{cases} P(u) & \text{if } U \text{ is not instantiated} \\ 0 & \text{if } U \text{ is instantiated but} \\ & \text{not to the value } u \\ 1 & \text{if } U \text{ is instantiated} \\ & \text{to the value } u \end{cases} \tag{7}$$

Step 2: Belief updating. The node X updates its belief measure to

$$Bel(x) = \alpha\lambda(x)\pi(x) \tag{8}$$

where

$$\lambda(x) = \begin{cases} \prod_j \lambda_{V_j}(x) & \text{if } X \text{ is not instantiated} \\ 0 & \text{if } X \text{ is instantiated but} \\ & \text{not to the value } x \\ 1 & \text{if } X \text{ is instantiated} \\ & \text{to the value } x \end{cases} \tag{9}$$

is the λ value of node X, and

$$\pi(x) = \begin{cases} \sum_{u_1,\ldots,u_n} & P(x|u_1,\ldots,u_n)\prod_i \pi_x(u_i) \\ & \text{if } X \text{ is not instantiated} \\ 0 & \text{if } X \text{ is instantiated but} \\ & \text{not to the value } x \\ 1 & \text{if } X \text{ is instantiated} \\ & \text{to the value } x \end{cases} \quad (10)$$

is the π value of node X, and where α is a normalizing constant.

*Step 3:*Bottom-up propagation. The node X computes new λ messages and posts them to its patents:

$$\lambda_X(u_i) = \sum_x \lambda(x) \sum_{u_k:k\neq i} p(x|u_1,\ldots,u_n)\prod_{k\neq i}\pi(u_k) \quad (11)$$

*Step 4:*Top-down propagation. The node X computes new π messages and posts them to its children:

$$\pi_{V_j}(x) = \begin{cases} Bel(x)/\lambda_{V_j}(x) & \text{if } X \text{ is not instantiated} \\ 0 & \text{if } X \text{ is instantiated but} \\ & \text{not to the value } x \\ 1 & \text{if } X \text{ is instantiated} \\ & \text{to the value } x \end{cases} \quad (12)$$

References

[1] L. N. Cassell, C. Partridge, Network management architectures and Protocols: problems and approaches. *IEEE Journal on Selected Areas in Communications*, Vol. 7, pp1104-1114 (1989)

[2] F. R. K. Chung, Reliable software and communication I: an overview. *IEEE Journal on Selected Areas in Communications*, Vol. 12, pp23-32, (1994)

[3] R. H. Deng, A. A. Lazar. A Probabilistic approach to fault diagnosis in linear lightwave networks. In *IEEE Journal On Selected Areas In Communication Networks*, Vol 11, No 9, pp: 1438-1448, (1993)

[4] Cynthia S. Hood and Chuanyi Ji. Intelligent Processing Agents For Network Fault Detection. IEEE Internet Computing (1997)

[5] M. Frontini, J. Griffin, and S. Towers. A Knowledge-Based System For Fault Localization in Wide Area Networks. In *Integrated Network Management, II,* pp 519-530. North Holland, (1991)

[6] R. Boubour. Suivie de panne par corrélation causale d'alarmes dans les systèmes réparties: Application aux réseaux de télécommunication. PhD dissertation. Rennes university, (1997)

[7] Irene Katzela and Mischa Schwartz. Schemes for fault identification in communication networks. In *IEEE/ACM Transactions on Networking*, Volume 3, No. 6 pp 753-764,(1995)

[8] Kalagnanam J. and Henrion M. A comparison of decision analysis an expert rules for sequential diagnosis. In *Proceedings of the fourth workshop on uncertainty in artificial intelligence*, (1988)

[9] Heckerman D., Breese J. and Rommelse K. Decision-theoretic troubleshooting. In *Communications of the ACM*, 38(3), pp49-57, (1995)

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Revised Edition, Morgan Kaufmamm Publishers. (1991)

[11] Yang Xiang. A probabilistic framework for multi-agent distributed interpretation and optimization of communication. *Artificial Intelligence*, Vol.87, No.1-2, pp 295-342, (1996).

[12] J. Pearl. Fusion, propagation and structuring in belief networks. *Artificial Intelligence* 29, pp 241-288, (1986)

[13] Cooper G. F. The Computational complexity of probabilistic Inference using Bayesian Belief networks is NP-Hard. *Artificial intelligence* 42: 393-405. (1990)

[14] Finn V. Jensen, S. L. Lauritzen, and K. G. Oleson. Bayesian updating in causal probabilistic networks by local computations. In *Comput. Stat. Quarterly*, (4) : pp 269-282, (1990). (Also in *An introduction to Bayesian networks.* UCL Press (1996))

[15] Stephan K. and Kurt G. A generic model for fault isolation in integrated management systems. *Journal of Network and Systems Management*,Special issue on fault management, 2nd quarter, (1997)