Predictive fault management in the dynamic environment of IP networks

Conference Paper · November 2004
DOI: 10.1109/IPOM.2004.1547622 · Source: IEEE Xplore

CITATIONS
13

5 authors, including:

Bernd J. Krämer
FernUniversität in Hagen
304 PUBLICATIONS **1,511** CITATIONS

SEE PROFILE

Shihao xu
University College Cork
2 PUBLICATIONS **14** CITATIONS

SEE PROFILE

Predictive Fault Management in the Dynamic Environment of IP Networks

Jianguo Ding*†, Bernd Krämer†, Shihao Xu*, Hansheng Chen§ and Yingcai Bai*
*Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai 200030, P. R. China
Email: Jianguo.Ding@sjtu.edu.cn
†Department of Electrical Engineering and Information Engineering
FernUniversität Hagen, D-58084, Germany
Email: Bernd.Kraemer@FernUni-Hagen.de
‡East-china Institute of Computer Technology, Shanghai 200233, P. R. China

***Abstract*— The growing complexity of IP networks in terms of hardware components, operating system, communication and application software and the huge amount of dependencies among them have caused an increase in demand for network management systems, particularly in fault management. An efficient fault detection system needs to work effectively even in face of incomplete management information, uncertain situations and dynamic changes. In this paper, dynamic Bayesian networks are proposed to model static and dynamic dependencies between managed objects in IP networks. Prediction strategies and a backward inference approach are provided for the proactive management in fault detection based on the dynamic changes of IP networks.**

I. Introduction

As IP networks grow in size, heterogeneity, pervasiveness, and complexity of applications and network services, their effective management becomes more important and more difficult. Managers have to live with unstable, uncertain, incomplete management information and continuous updates of network systems. Individual hardware defects or software errors or combinations thereof occurring in different system components may cause the degradation of services of other (remote) components in the network or even their complete failure due to functional dependencies between managed objects. Hence an effective distributed fault detection mechanism is needed to support rapid decision making in detecting and diagnosing faults at an early stage and allow for partial automation of fault correction. In the past decade, a great deal of research effort has been focused on improving management systems in fault detection and diagnosis. Rule-based expert systems have been so far the major approach for solving the alarm correlation problem [17], [24], [27]. This approach suits well-defined problems where the environment is not very dynamic. Using Finite State Machines (FSMs) to detect faults is another approach [2], [13], [21] to cope with incomplete information and unforeseen faults. Case-based reasoning offers potential solutions to the problems of adaptation and knowledge acquisition bottlenecks [12], [19]. However, most of these solutions are very sensitive to ”noise” (such as loss, delay, corruption of messages etc.) and are unable to deal with incomplete and imprecise management information effectively. Probabilistic reasoning is an effective approach for fault detection in network management [6], [10], [22], [23].

On the practical side, most of the current commercial management software, such as IBM Tivoli, HP OpenView, or Cisco serial network management software, support the integration of different management domains, collect information, perform remote monitoring, provide fault alarm, and perform some statistics on management information. But they still lack facilities to adapt to dynamic changes in network systems for exact fault localization, or to the automatic execution of appropriate fault recovery actions. From the experiences in network management including most commercial management software, a typical measure for on-line fault identification is 95% fault location accuracy but 5% faults cannot be located and recovered in due time [7]. Dynamic changes raise even higher barriers for exact fault location. Hence, for large IP networks including thousands of managed components it may be rather time-consuming and difficult to locate the unknown cause of faults in due time by exhaustive search for the root causes of a failure and this process may interrupt or impair important system services. Dynamic updates bring up even more challenges in fault detection.

In this paper we apply Dynamic Bayesian Networks (DBNs) to address temporal factors and model dynamic changes of managed entities and the dependencies between them. Based on related inference techniques we further investigate prediction capabilities in fault management in the presence of imprecise and dynamic management information.

The application of DBNs to fault management is discussed in Section II. The prediction mechanisms reflecting dynamic changes in IP networks are presented in Section III. Simulation results are presented in Section IV. Section V concludes and identifies directions for further research.

II. Dynamic Bayesian Networks for IP Network Management

A. Dynamic Characteristics in IP networks

Dynamic updates in network systems can be classified as either hard or soft changes. A hard change means that the change happens abruptly and most of the time a hard-change

0-7803-8836-4/04/$20.00 ©2004 IEEE.

is generated on purpose by the system owner. This kind of change does not depend on the system's history. For example, a router being added or removed from the network may cause an abrupt change in the system topology and behavior. Some intended operations can also generate this kind of hard change, such as a change of the configuration of a network system. Generally, a hard change does not happen so often, and the changes could be predictable based on the intention of the system manager. While soft changes mean that a change happens gradually and depends on the system history. a soft change typically results from changes of system properties such as performance degradation, application degeneration, or dependency modification. From the experience of systems management, lots of unknown or un-located causes of faults are triggered by a soft change, which is related to the potential changes and updates of the system. Compared with a hard change, a soft-change keeps going on all the time in IP networks and it is hard to predict by straight-forward approach. In our research we focus on soft changes.

If we consider soft changes in IP networks, one kind of change comes from individual network entities, another arises from updating dependencies between managed entities. From the viewpoint of management, an entity can be a hardware device, a software component or a certain application.

In computer systems and IP networks, real-life dynamic systems are rife with nonlinearities, many of which are expressed as discrete failure modes that can produce discontinuous jumps in system behavior.

When a dynamic system is modeled, a time dimension will be considered. Because observations and evidences can be updated over time, the management system should capture the evolution of the system as it changes over time.

B. Standard and Dynamic Bayesian Networks

1) Bayesian Networks: BNs [3], [18] use DAGs (Directed Acyclic Graphs) with probability labels to represent probabilistic knowledge. BNs can be defined as a triplet (V, L, P), where V is a set of variables (nodes of the DAG), L is the set of causal links among the variables (the directed arcs between nodes of the DAG), P is a set of probability distributions defined by: $P = \{p(v \mid \pi(v))|v \in V\}$; $\pi(v)$ denotes the parents of node v. The DAG is commonly referred to as the dependence structure of a BN.

An important advantage of BNs is the avoidance of building huge joint probability distribution tables that include permutations of all the nodes in the network. Rather, for an effect node, only the states of its immediate predecessor need to be considered.

Fig.1 shows a particular detail of the campus network of the FernUniversität in Hagen.

When only the connection service for end users is considered, Fig. 2 can be the correlated BN. The arrowed lines in BN denote the dependency from causes to effects. The annotation $p(\overline{D}|\overline{E}F) = 100\%$ denotes the probability of non-connectivity for component D is 100% when component F is in order but the connection of component E is not available.

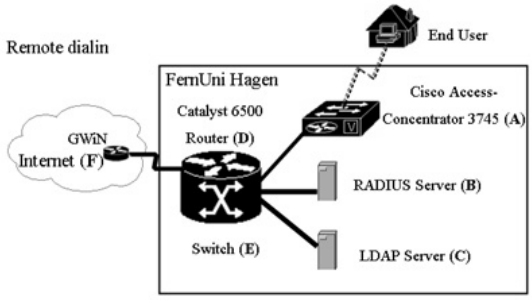
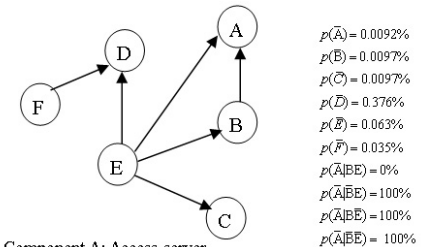


Fig. 1. Example of Campus Network



Component A: Access-server
Component B: RADIUS-Server
Component C: LDAP-SERVER
Component D: Router- Connection to Internet-Provider
Component E: Switch for LAN Access
Component F: GWiN and Internet Access

$p(\overline{A}) = 0.0092\%$
$p(\overline{B}) = 0.0097\%$
$p(\overline{C}) = 0.0097\%$
$p(\overline{D}) = 0.376\%$
$p(\overline{E}) = 0.063\%$
$p(\overline{F}) = 0.035\%$
$p(\overline{A}|BE) = 0\%$
$p(\overline{A}|\overline{B}E) = 100\%$
$p(\overline{A}|B\overline{E}) = 100\%$
$p(\overline{A}|\overline{B}\overline{E}) = 100\%$
$p(\overline{D}|EF) = 0\%$
$p(\overline{D}|\overline{E}F) = 100\%$
$p(\overline{D}|E\overline{F}) = 20\%$
$p(\overline{D}|\overline{E}\overline{F}) = 100\%$
$p(\overline{B}|\overline{E}) = 100\%$
$p(\overline{B}|E) = 0\%$
$p(\overline{C}|\overline{E}) = 100\%$
$p(\overline{C}|E) = 0\%$

Fig. 2. Example of Bayesian Network for Fig. 1

Other annotations can be explained by similar way. In Fig. 2, some evidences, such as the status of component D are easy to detect from the view of management, but the causes of component D are not obvious to be observed. One important task in management is to infer the hidden factors from the available evidences.

2) Dynamic Bayesian Networks: A problem with the standard theory of BNs is that there is no natural mechanism for representing time [1], [26].

DBNs model a system that is dynamically evolving over time [9], [16], [20]. This model will enable the user to monitor and update the system as time proceeds. As there is no standard definition for DBNs, researchers may use different descriptions to accommodate their research requirements. The literature tends to use the terms "dynamic" and "temporal" interchangeably.

The temporal approaches can be divided into two main categories of time representation, namely those models which represent time (1) as points or instances or (2) as time intervals.

DBNs can be considered as time related function: $BN(t) = (V(t), L(t), P(t))$. For a soft change, dynamic changes only happen in individual components and on the dependency between components. Under this kind of changes, the topology of the BN keeps stable, hence the time parameter can be omitted in nodes and edges: $BN(t) = (V, L, P(t))$.

Due to the dense knowledge representation of DBNs, DBNs

can represent large amounts of interconnected and causally linked data as they occur in IP networks. Generally speaking: (1) DBNs can represent knowledge in depth by modeling the functionality of the transmission network in terms of cause and effect relationships among network components and network services. (2) They can provide guidance in diagnosis. Calculations over a BN can determine both the precedence of detected effects and the network part that needs further investigation in order to provide a finer grained diagnosis. (3) They have the capability of handling uncertain and incomplete and dynamic management information due to their grounding in probability theory and nonlinear regression. (4) They provide a compact and well-defined problem space since they use an exact solution method for any combination of evidence or set of faults.

C. Mapping Managed Networks to DBNs

An IP network consists of a number of managed objects. An object is a 'part' of the network system that has a separate and distinct existence. An object can be a network, a node, a switch, a layer in a protocol stack, a software process, a virtual link, a physical element like an optical fiber, a piece of cable, a hardware component, etc. The concept of division and appropriate level of division are system and application dependent.

Objects in an IP network are dependent upon each other in rather complex ways. These dependencies are very important for the alarm correlation and fault identification process. Most of the time a failure in one object has side effects on other objects that are dependent on it. For example, a link failure has an effect on other resources in the network, e.g., connections on the various layers which use the link will experience timeouts. The knowledge of these dependencies gives us valuable information for the purpose of alarm correlation and fault localization.

Uncertainty about dependencies among network entities is represented by assigning probabilities to the links in the dependency or causality graph [10], [11]. Some commonly accepted assumptions in this context are that (1) given fault a, the occurrences of faults b and c that may be caused by a are independent, (2) given occurrence of faults a and b that may cause event c, whether a actually causes c is independent of whether b causes c (the OR relationship among alternative causes of the same event), and (3) root faults are independent of one another. We take advantage of these approximating assumptions throughout the paper.

When an IP network is modeled as a DBN, two important processes need to be resolved:

1) Ascertain the Dependency Relationship and its Changes between Managed Entities: Dependencies represent service dependency relationships between various cooperating entities in a network. When one entity requires a service performed by another entity in order for it to execute its function, this relationship between the two entities is called a dependency. When a managed entity A (such as a service, an application component in software or hardware) depends on a managed entity B, we say that A is the dependent and B is the antecedent. The notion of dependencies can be applied at various levels of granularity. Sometimes the dependencies that occur between different system components should be defined carefully. For example, the maintenance of an Email server obviously affects the service 'Email' and thus all the users whose user agents have a client/server relationship with this specific server; however, other services (News, WWW, FTP) are still usable because they do not depend on a functioning Email service. So the inter-system dependencies are always confined to the components of the same service.

Two models are useful to get the dependency between cooperating entities in IP networks.

The functional model defines generic service dependencies and establishes the principle constraints to which the other models are bound. A functional dependency is an association between two entities, typically captured first at design time, which says that one component requires some services from another.

The structural model contains the detailed description of software and hardware components that realize the service. A structural dependency contains detailed information and is typically captured first at deployment or installation time.

The timed period recording of the dynamic changes in the dependencies is the important source of analysis and evaluation of the network system performance.

2) Measuring Dependencies: When BNs are used to model IP networks, BNs represent causes and effects between observable symptoms and the unobserved problems, so that when a set of evidences is observed the most likely causes can be determined by inference technologies. Single-cause and multi-cause faults are two kinds of general assumptions to consider the dependencies between managed entities in IP networks management. A non-root node may have one or several parents (causal nodes). Single-cause means any of the causes must lead to the effect. While multi-cause means that some effect is generated only when more than one cause happens simultaneously. Past management information statistics are the main source to get the dependencies between the managed objects in IP networks. The empirical knowledge of experts and experiments are useful to determine the dependencies.

Some researchers have performed useful work to discover dependencies from the application view in networks [5], [8].

Time series of dynamic dependency provides the possibility for further prediction in fault management and performance evaluation.

III. Prediction Strategies in DBNs for IP Networks

Correlation serves to diminish the number of alarms presented to the operator in network management, yet ideally the approach should be able to facilitate fault prediction, which can predict the fault(s) that have occurred from the alarms and warn the operator before severe faults may happen.

Considering the model of DBNs in Fig. 3, two possible changes, which are updates over time, are presented in DBNs:

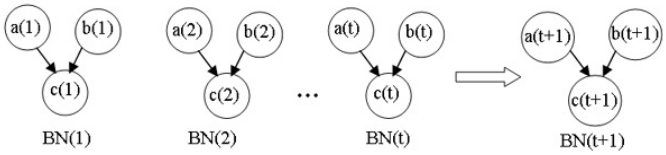


Fig. 3. Model of Dynamic Bayesian Network

(1) the probability update in the nodes (variables) and (2) the probability updates in links (dependency between nodes).

When a network is modeled as DBNs, one important task is to capture the trends for the evolution in the network. This amounts to obtain $BN(t+1)$ based on the data set $BN(0), BN(1), \ldots, BN(t)$. Here $BN(t)$ denotes the update BN at time t.

In DBNs, the following prediction tasks are to be considered in face of the management requirement.

(1) Prediction per individual component. The state of an individual component in an IP network can change over time because of the degradation or improvement of the component. The prediction of the individual component's state change can be denoted as: $p(v(1)), p(v(2)), \ldots, p(v(t)) \rightarrow p(v(t+1)), v \in V$. $p(v(t))$ represents the probability of the state of component v at time t.

(2) Prediction of the dependency relationship between components. The modification of dependencies between managed objects derives from updating the system performance and changes in the correlation between objects. This can be denoted as: $p(v(1)|\pi(v(1)))$, $p(v(2)|\pi(v(2)))$, $\ldots$, $p(v(t)|\pi(v(t))) \rightarrow p(v(t+1)|\pi(v(t+1)))$, $v \in V$. $p(v(t)|\pi(v(t)))$ represents the probability of the dependency between node v and its parent $\pi(v)$ at time t.

(3) Prediction for potential faults based on backward inference. When the future state of the effect nodes is estimated, a promising prediction is to trace the causal nodes based on the estimated state of the effect nodes. The prediction from effects to causes is considered as the backward inference: $E(t+1) \rightarrow C(t+1)$. $E(t)$ denotes the set of effects at time t, and $C(t)$ denotes the set of causes at time t.

A. Single Factor (Variable) Prediction in DBNs

Since the dynamic changes in IP networks are identified as nonlinear time series, the Least Squares Fit (LSF) of a polynomial is applied for nonlinear regression [25].

Suppose that we want to fit a polynomial $y = a_0 + a_1 x + \ldots + a_m x^m$ with the data points $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)$. In (x_i, y_i), x_i denotes the time dimension; y_i denotes the time-series data. Then we have:

$$\begin{cases} y_1 = a_0 + a_1 x_1 + + a_m x_1^m \\ y_2 = a_0 + a_1 x_2 + + a_m x_2^m \\ \qquad \ldots \\ y_n = a_0 + a_1 x_n + + a_m x_n^m \end{cases}$$

We use a matrix to model the equations above: $y = Av$, $y = [\ y_1 \quad y_2 \quad \ldots \quad y_n\ ]^T$, $A = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^m \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ 1 & x_3 & x_3^2 & \ldots & x_3^m \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 1 & x_n & x_n^2 & \ldots & x_n^m \end{pmatrix}$, $v = [\ a_0 \quad a_1 \quad \ldots \quad a_m\ ]^T$. Then $A^T A v = A^T y$, hence $v = (A^T A)^{-1} A^T y$.

Week No.	Failure rate (%)	Week No.	Failure rate (%)	Week No.	Failure rate (%)	Week No.	Failure rate (%)	Week No.	Failure rate (%)
1	0.000	12	0.000	23	0.000	34	0.298	45	0.000
2	6.860	13	0.000	24	0.000	35	0.000	46	0.000
3	0.000	14	0.000	25	6.220	36	1.617	47	0.000
4	0.000	15	0.000	26	0.030	37	0.000	48	0.050
5	0.000	16	0.000	27	1.964	38	0.000	49	0.000
6	0.000	17	0.000	28	0.000	39	0.000	50	0.000
7	0.000	18	0.000	29	0.893	40	0.000	51	0.000
8	0.000	19	0.000	30	1.081	41	0.000		
9	0.000	20	0.000	31	0.000	42	0.000		
10	0.248	21	0.000	32	0.000	43	0.000		
11	0.000	22	0.000	33	0.000	44	0.000		

Fig. 4. The Data Set of Failure Ratio of Component D in Fig. 2

From the calculation above, the coefficient of the polynomial $y = a_0 + a_1 x + \ldots + a_m x^m$ is obtained. The prediction of the value y can be obtained given x (time variable).

As an example, consider the dynamic changes in the failure time on component D illustrated in Fig. 2. We use the statistics recorded in the system log to calculate the failure time in component D. The weekly time related failure state in component D is presented in Fig. 4.

For simplification, we use fourth order polynomial to perform the nonlinear regression:

$Y = A + B * X + C * X^2 + D * X^3 + E * X^4$

Based on the data set shown in figure 4, the best-fit values are obtained: $A = 3.046, B = -0.7170, C = 0.05246, D = -0.001407, E = 1.2430e-005$.

Now the value of the failure state for week 52 in component D can be predicted by: $Y = A + B * X + C * X^2 + D * X^3 + E * X^4 = 3.046 - 0.7170 * 52 + 0.05246 * 52^2 - 0.001407 * 52^3 + (1.2430e-005) * 52^4 = 0.6648$.

Hence in week 52, the estimated value of the non-availability time in component D is $0.6648\% * 10080$ (minutes for 7 days)=67.01 (minutes). From the record of the system log, the failure time in week 52 is 58 minutes. The reason for the deviation between predicted and observed down-time are some errors in the prediction. These prediction errors can be used to correct the future estimation. From the practical point of view, the alerts or reminders from the management system provide the manager helpful references for better system protection, when the estimation of the failure time exceeds a threshold.

The nonlinear prediction for the dynamic updating of dependencies in DBNs can be processed by the same procedure.

In real-life IP networks, dynamic changes may not spread on each object and each dependency link in a management period. Most of the time only partial updates are on going

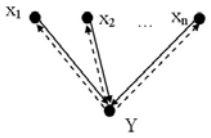


Fig. 5. Basic Model for Backward Inference in BNs

indeed, or else the system would be totally unstable and it is hard to provide applicable predictions of system behavior.

The precision of the prediction in nonlinear time series is related to the size of the data set. The prediction will be more precise if larger data sets are available. Hence the prediction is advisable for the long term behavior of networks. Meanwhile the noise in the data is another factor which affects the result of prediction.

B. The Prediction Based on Probabilistic Inferences in IP Networks

The most common approach towards reasoning with uncertain information about dependencies in networks is probabilistic inference, which traces the causes from effects. The task of backward inference amounts to finding the most probable instances of some hypothesis variables, given the observed evidence.

Suppose the final estimated values, which come from the dynamic changes in the whole DBN, have already been obtained from the nonlinear regression, which is stated in section 3. Now consider the backward inference in static BNs. E is defined as the set of effects (evidences) which we can observe, and C as the set of causes.

Fig. 5 shows the basic model for backward inference in BNs. Let $X = (x_1, x_2, \ldots, x_n)$ be the set of causes. According to the definition of BNs, the following variables are known: $p(x_1), p(x_2), \ldots, p(x_n), p(Y|x_1, x_2, \ldots, x_n) = p(Y|X)$. Here $x_1, x_2, \ldots, x_n$ are mutually independent, so

$$p(X) = p(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} p(x_i) \tag{1}$$

$$p(Y) = \sum_{X}[p(Y|X)p(X)] = \sum_{X}[p(Y|X)\prod_{i=1}^{n} p(x_i)] \tag{2}$$

by Bayes' theorem,

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)} = \frac{p(Y|X)\prod_{i=1}^{n} p(x_i)}{\sum_{X}[p(Y|X)\prod_{i=1}^{n} p(x_i)]} \tag{3}$$

which computes to

$$p(x_i|Y) = \sum_{X\backslash x_i} p(X|Y) \tag{4}$$

In Eq. (4), $X\backslash x_i = X - \{x_i\}$. According to Eqs. (1)-(4), the individual conditional probability $p(x_i|Y)$ can be achieved from the JPD $p(Y|X)$, $X = (x_1, x_2, \ldots, x_n)$. The backward dependency can be obtained from Eq. (4). The dashed arrowed lines in Fig.5 denote the backward inference from effect Y to individual cause $x_i, i \in [1, 2, \ldots, n]$.

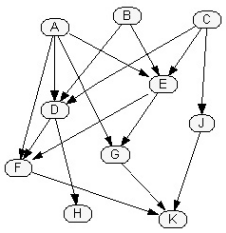


Fig. 6. Simulation of Dynamic Bayesian Network

In Fig. 2, when a fault in component D (connection to Internet Provider) is detected, the state of D would be non-connectivity, say $p(\overline{D}) = 1$. Then based on Eqs. (1)-(4), we obtain $p(\overline{F}|\overline{D}) = 67.6\%, p(\overline{E}|\overline{D}) = 32.4\%$. This can be interpreted as follows: when component D is not available, the probability of a fault in component F is 67.6% and the probability of a fault in component E is 32.4%. Hence component F is more likely to be the cause of the failure in component D. Here only the fault related to connection service is considered.

From the viewpoint of an application, however, a BN will be multi-leveled in topology. Tracing the strongest dependency route and strongest causes are important management task in IP network management. Our previous work, the Strong Dependency Route (SDR) algorithm which is used for backward inference in complex BNs, is presented in [4].

Consider the backward inference in DBNs, the dynamic changes in individual nodes or individual dependencies may propagate to the whole DBN and thus cause the modification of the strongest dependency routes and the rank of the dependency sequence in causal nodes. The simulation result in section IV shows more details of this kind of dynamic changes in DBNs.

IV. Simulation Measurement in DBNs

For the simulation in DBNs, a robust and reliable random number generator plays an important part. Based on the TGFSR (Twisted Generalized Feedback Shift Register) algorithm [14], [15] for the random number generation, we develop a simulator. The simulator can generate the BNs and the data set for inference test.

Fig. 6 presents the topology of the simulated Bayesian network. The cause set $C = \{A, B, C\}$, the effect set $E = \{H, K\}$. The JPD which describes the Bayesian network is denoted in Fig. 7.

(1) Firstly, consider the backward inference in static Bayesian networks. Based on the SDR algorithm [DKB+04], a spanning tree, which holds the strong dependency routes, is obtained (see Fig. 8).

From the spanning tree, the strongest routes between effects and causes can be obtained by depth-first search. Meanwhile the inference also provides a cause sequence, in which the

P(A=1) = 0.007730	P(F=1|E=0,D=0,A=0) = 0.000000
P(B=1) = 0.014476	P(F=1|E=0,D=0,A=1) = 0.328807
P(C=1) = 0.071084	P(F=1|E=0,D=1,A=0) = 0.553357
P(D=1|C=0,A=0,B=0) = 0.000000	P(F=1|E=0,D=1,A=1) = 0.761124
P(D=1|C=0,A=0,B=1) = 0.734824	P(F=1|E=1,D=0,A=0) = 0.768399
P(D=1|C=0,A=1,B=0) = 0.751251	P(F=1|E=1,D=0,A=1) = 0.506738
P(D=1|C=0,A=1,B=1) = 0.855490	P(F=1|E=1,D=1,A=0) = 0.700524
P(D=1|C=1,A=0,B=0) = 0.420072	P(F=1|E=1,D=1,A=1) = 0.947701
P(D=1|C=1,A=0,B=1) = 0.806797	P(K=1|G=0,J=0,F=0) = 0.000000
P(D=1|C=1,A=1,B=0) = 0.559901	P(K=1|G=0,J=0,F=1) = 0.221167
P(D=1|C=1,A=1,B=1) = 0.922359	P(K=1|G=0,J=1,F=0) = 0.335382
P(G=1|E=0,D=0) = 0.000000	P(K=1|G=0,J=1,F=1) = 0.402651
P(G=1|E=0,D=1) = 0.575105	P(K=1|G=1,J=0,F=0) = 0.082212
P(G=1|E=1,D=0) = 0.335308	P(K=1|G=1,J=0,F=1) = 0.389683
P(G=1|E=1,D=1) = 0.712756	P(K=1|G=1,J=1,F=0) = 0.470210
P(E=1|B=0,C=0,A=0) = 0.000000	P(K=1|G=1,J=1,F=1) = 0.799994
P(E=1|B=0,C=0,A=1) = 0.438063	P(H=1|D=0) = 0.000000
P(E=1|B=0,C=1,A=0) = 0.590781	P(H=1|D=1) = 0.885234
P(E=1|B=0,C=1,A=1) = 0.629491	P(J=1|C=0) = 0.000000
P(E=1|B=1,C=0,A=0) = 0.295333	P(J=1|C=1) = 0.137611
P(E=1|B=1,C=0,A=1) = 0.588910	
P(E=1|B=1,C=1,A=0) = 0.834129	
P(E=1|B=1,C=1,A=1) = 0.921826	

Fig. 7. The JPD of BN in Fig. 6 (here 0 denotes normal state, 1 denotes abnormal state)

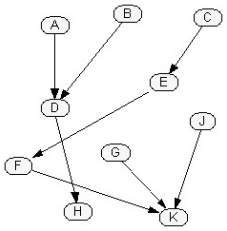


Fig. 8. The Spanning Tree from Static BN in Fig. 6

strongest route between effect nodes and cause nodes can be achieved too.

In the simulation experiment, detection rate represents the percentage of faults that occurred in the network in a given experiment that were detected by an algorithm. As stated in Section 1, when an application system meets the 5% non-located faults, generally the manager has to detect them randomly or exhaustively. The exhaustive detection will be rather time-consuming and intractable when the IP network is large and complex enough.

The objective of any fault management system is to minimize the time to locate a fault. This time is the sum of the time to propose possible fault hypotheses (fault identification) and the time to perform testing in order to verify these hypotheses. The time required for testing is affected by the number of managed objects that must be tested. Thus, if the network management system is able to identify the source of a fault, it is desirable that the minimum number of tests be performed. Hence, there are two main aspects, subject to optimization, of any fault localization process: accuracy of the hypothesis it provides and time complexity of the fault identification algorithm it uses. In order to optimize the time to locate the fault, we should maximize the accuracy of the proposed

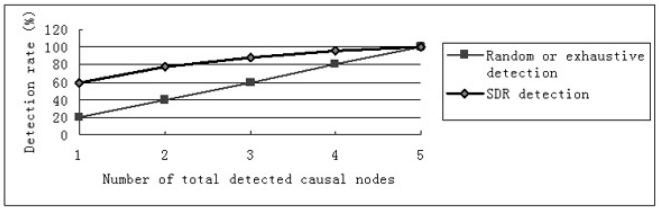


Fig. 9. Comparison of the Detection Rate between Random Detection and SDR Detection in Static BN

Time interval (t)	1	2	3	4	5	6	7	8	9
p(A=1)	0.007	0.109	0.230	0.410	0.533	0.601	0.712	0.799	0.856

Fig. 10. The Updating States in Node A of Fig. 6

hypotheses and minimize the time complexity of the fault identification process.

During the 10000 tests, the simulation results are presented in Fig. 9.

(2) Now, consider the dynamic changes in the BN. For simplification, in Fig. 6, suppose the changes only happen on node A, while the other node and links keep stable. The updated states of A are denoted in Fig. 10.

Based on the non-linear regression, at time interval $t = 10$, $p(A = 1) = 0.972$. Then, through the SDR algorithm, the spanning tree is obtained (see Fig. 11). The strongest dependency routes are modified with the state updates in node A. The comparison of the detection rate between the inference based on DBNs and random detection is presented in Fig. 12.

In backward inference, when the threshold of the failure alert is set to $1/n$, all the causal nodes which hold the probability greater than $1/n$ are considered as critical cause nodes. From the simulation result, more than 80% cause nodes can be detected by only checking the critical causal nodes (less

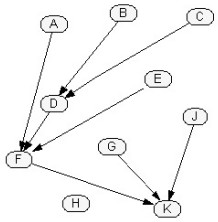


Fig. 11. The Spanning Tree of BN in Fig. 6 after the Dynamic Changes in Node A

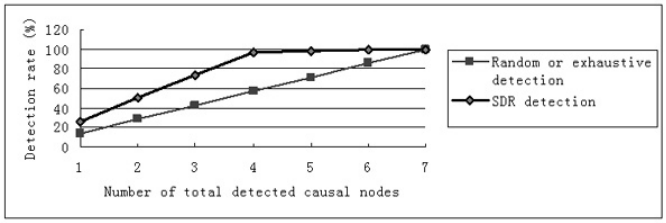


Fig. 12. Comparison of the Detection Rate between Random Detection and SDR Detection in DBN

than 50% of the whole cause nodes). Comparing this with the random or exhaustive detection, the backward inference in DBNs provides a more efficient approach to catch the causal nodes. When the set of causal nodes is larger, the detection rate is more optimal.

V. Conclusions and Future Works

In IP networks of realistic size and complexity, managers have to face the unstable, uncertain and incomplete information and dynamic updates of management objects and their dependencies among each other. We used BNs to represent the knowledge about managed objects and their dependencies and apply probabilistic reasoning to determine the causes of failures or errors. In order to model the dynamic changes in a network system, temporal extensions of BNs are employed to integrate the time dimension and further to present the prediction strategies based on backward inference in fault management in dynamic environment of network systems. Not only the fault prediction in an individual entity and the dependency relationship between managed entities are considered, but also the potential fault detection from effects to causes in DBNs are investigated. Particularly when the management system can not give exact fault location, the related inference and prediction approaches can act as an effective mechanism in diagnosis, ranking possible failures, handling of multiple simultaneous failures, and robustness by probabilistic reasoning and demonstrate a more optimal fault detection rate than random detection or exhaustive detection.

Dynamic changes in IP networks may cause topology changes in managed networks and vibration in system performance. The JPD in DBNs contains the knowledge of system updating. In our future work, we try to develop a heuristic method to identify the changes in topology and structure and evaluate the performance of dynamic network systems. The entropy theory will be applied to DBNs and its application in performance management will be investigated.

Acknowledgment

This research is part of the IQN (International Quality Networks) project and was supported by the DAAD (the German Academic Exchange Service). The authors would like to thank Carsten Schippang for providing the sample data of the campus network of FernUniversität in Hagen for a whole year. Also many thanks are due to the anonymous reviewers for their valuable comments.

References

[1] C. F. Aliferis, G. F. Cooper. A Structurally and Temporally Extended Bayesian Belief Network Model: Definitions, Properties, and Modeling Techniques. Proc. of the 12th Conference on Uncertainty in Artificial Intelligence, pp.28-39, 1996.

[2] CS Chao, DL Yang, AC Liu. A LAN fault diagnosis system. Computer Communications. Vol24, pp1439-1451, 2001.

[3] R. Dechter. Bucket Elimination: A unifying framework for probabilistic inference. Proc. of the 12th Conference on Uncertainty in Artificial Intelligence, Portland, Oregon, Morgan Kaufmann Publishers, Aug. 1996.

[4] J. Ding, B. Krämer, Y. Bai, H. Chen. Probabilistic Inference for Network Management. Universal Multiservice Networks, M. Freire et al. (Eds.). Lecture Notes in Computer Science (LNCS: 3262) ISBN: 3-540-23551-5, pp. 498–507, Springer-Verlag Berlin Heidelberg, 2004.

[5] M. Gupta, A. Neogi, M. K. Agarwal and G. Kar. Discovering Dynamic Dependencies in Enterprise Environments for Problem Determination. Proc. of 14th IEEE/IPIP International Workshop on Distributed Systems Operations and Management. Heidelberg, Germany, 2003.

[6] C. S. Hood and C. Ji. Proactive network-fault detection. IEEE Transactions on Reliability, 46(3):333–341, September 1997.

[7] C. Hill. High-availability systems boost network uptime: Part 1. http://www.eetasia.com/ARTICLES/2001JUL/2001JUL01_NTEK_ST_QA_TA.PDF. Motorola Telecom Business Unit, 2001.

[8] A. Keller, U. Blumenthal, G. Kar. Classification and Computation of Dependencies for Distributed Management. Proceedings of 5th IEEE Symposium on Computers and Communications. Antibes-Juan-les-Pins, France, July 2000.

[9] K. Kanazawa, D. Koller, S. Russell, "Stochastic Simulation Algorithms for Dynamic Probabilistic Networks", Proc. of the 11th Annual Conference on Uncertainty and Artificial Intelligence, 1995.

[10] I. Katzela and M. Schwartz. Schemes for fault identification in communication networks. IEEE Transactions on Networking, 3(6):733-764, 1995.

[11] S. Klinger , S. Yemini , Y. Yemini , D. Ohsie , S. Stolfo. A coding approach to event correlation. Proceedings of the fourth international symposium on Integrated network management IV, pp.266-277, January 1995.

[12] L. Lewis. A case-based reasoning approach to the resolution of faults in communication networks. In Integrated Network Management, III, 671-682. Elsevier Science Publishers B.V., Amsterdam, 1993.

[13] R.E. Miller, K.A. Arisha. Fault management using passive testing for mobile IPv6 networks. Proc. of 2001 IEEE Global Telecommunications Conference. Volume: 3, pp. 1923 -1927, 2001.

[14] M. Matsumoto and Y. Kurita. Twisted GFSR generators. ACM Trans. on Modeling and Computer Simulation, 2(1992), pp. 179-194, .

[15] M. Matsumoto and Y. Kurita. Twisted GFSR generatos II. ACM Trans. on Modeling and Computer Simulation, 4(1994)pp. 254-266.

[16] A. E. Nicholson, J. M. Brady. Dynamic Belief Networks for Discrete Monitoring. IEEE Trans. on Systems, Man and Cybernetics, Vol. 34(11), pp. 1593- 1610, 1994.

[17] A. Osmani, F. Krief, Model-based diagnosis for Fault Management in ATM Networks. Proc. of International Conference on ATM ICATM 99. pp. 91-99. June, 1999.

[18] J. Pearl. Causality: Models, Reasoning, and Inference. Cambridge, England: Cambridge University Press. New York, NY, ISBN: 0-521-77362-8, 2000.

[19] G. Pemido, J. Nogueira, and C. Machado. An automatic fault diagnosis and correction system for telecommunications management. Proc. of 6th IFIP/IEEE Int. Symp. Integrated Network Management, pp.777-791, 1999.

[20] S. Russell, J. Binder, D. Koller, K. Kanazawa. Local Learning in Probabilistic Networks with Hidden Variables. Proc. 14th Int. Joint Conf. on AI, pp.1146- 1152, 1995.

[21] I. Rouvellou and G. W. Hart. Automatic Alarm Correlation for Fault Identification. Proc. of IEEE INFOCOM'95, pp. 553-561, 1995.

[22] R. Sterritt, D. W. Bustard: Fusing hard and soft computing for fault management in telecommunications systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C 32(2): pp 92-98, 2002.

[23] M. Steinder and A. S. Sethi. Non-deterministic diagnosis of end-to-end service failures in a multi-layer communication system. In Proc. of ICCCN, pp. 374-379, Scottsdale, 2001.

[24] R. Vaarandi. Platform Independent Event Correlation Tool for Network Management. Proc. of the 2002 IEEE Workshop on IP Operations and Management. ISBN: 0-7803-7658-7.

[25] A. S. Weigend, and N. A. Gershenfeld. Time Series Prediction. Addison-Wesley, 1994.

[26] J. D. Young, E. Santos. Introduction to Temporal Bayesian Networks. Presented at the 7th Midwest AI and Cognitive Science Conf., 1996.

[27] J. Zupan, D. Medhi. An alarm management approach in the management of multi-layered networks. 3rd IEEE International Workshop on IP Operations & Management (IPOM 2003), pp 77-84. Oct. 2003