

# Fault Prediction at the Network Layer using Intelligent Agents

M. Thottan, C. Ji

Department of Electrical, Computer, and Systems Engineering  
Rensselaer Polytechnic Institute, Troy, NY 12180  
e-mail: thottm,chuanyi@ecse.rpi.edu

## Abstract

Prediction of network failures at the router level has been achieved using an Intelligent Agent. The intelligent agent gathers relevant MIB data from the router and provides temporally and spatially correlated predictive alarms. The time correlated abnormal changes in the individual MIB variables are spatially correlated using a novel combining scheme. The agent was implemented on a real network and seven out of nine faults were predicted. Two typical case studies are presented. The prediction time was in the order of minutes.

*Research Paper with Case Studies*

*Key Words:* intelligent agent, adaptive learning, MIB variables, Fault prediction

## I. INTRODUCTION

Routers form the primary nodes of a Wide Area Network(WAN). The ability to predict abnormal or faulty conditions at a router [13] is vital to providing Quality of Service (QoS) guarantees for real time services. Prediction of impending faults will allow for control measures to be taken which will result in traffic being routed away from a problem zone. The ability to predict router level failures could avert large scale network outages.

The challenge presented in the prediction of router level faults is to achieve accurate prediction with very low false alarm rate at the appropriate time scale. The occurrence of false alarms could add to instability in the network. Furthermore, since the agent is implemented at the router, the processing overhead due to the agent must be kept to a minimum. The agent implemented in this work addresses these issues and is capable of fault prediction.

Current commercial network management packages do not provide online fault prediction. Earlier work focussed on fault identification using fault models described by Finite State Machine models [16] [3] and using graph based identification techniques [12]. A review of network fault detection and identification can be found in [14]. As described in [11] these methods assumed that the alarms pertaining to fault events were provided along with accurate temporal information. However, the generation of predictive and reliable time correlated alarms still remained an open problem.

A new approach was proposed and implemented by Maxion and others [15] [7] which described faults as deviations from normal behavior. This method required feature vec-

Supported by DARPA under contract number F30602-97-C-0274 and the National Science Foundation ((CAREER) IRI-9502518)

tors which describe the faults. A promising approach employing Management Information Base (MIB) variables was introduced in our previous work [9]; a segmentation algorithm for online feature extraction and a combining scheme using Bayesian belief networks was implemented. As an improvement on this method we recently developed a change detection algorithm based on the Generalized Likelihood Ratio (GLR) test to generate feature vectors from a select set of MIB variables. These vectors were combined using a simple duration filter to get node level alarms [17][18].

In this work, an intelligent agent which provides temporally and spatially correlated predictive alarms was developed for the router. The new distributed architecture is scalable to any number of routers and is amenable for online implementation. The agent was implemented on real network data. Two case studies are presented to illustrate the capability of the agent. The approach used here provides a theoretical framework to the problem of fault prediction. The operator matrices introduced provide a geometric interpretation of the fault domain.

## II. INTELLIGENT AGENT: THE MODEL

The intelligent agent implemented at a router should pose minimal computational overload. Hence the agent was developed in a distributed framework as shown in Figure 1, where the agent uses the local MIB data to generate predictive alarms. Such a scheme was motivated by the work done on management by delegation [8]. The lo-

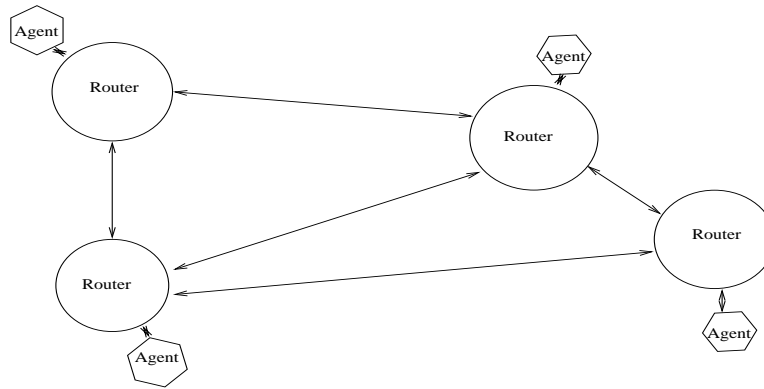


Fig. 1. Distributed Processing

cal processing done by the agent allows it to scale easily for any number of routers in the network. The information obtained at the router is the aggregate of the information from all the subnets. The router, which is primarily a network layer device, processes the *ip* layer information which is a multiplexing of traffic from all of the interfaces. This distributed scheme allows for problem isolation to a specific subnetwork.

The *Intelligent Agent* is a processing algorithm much like a software entity that has as its inputs the MIB variables that are specific to the router and its output provides a parameter that is a predictive indicator of network health.

The implementation of the agent consists of two stages as shown in Figure 2. The

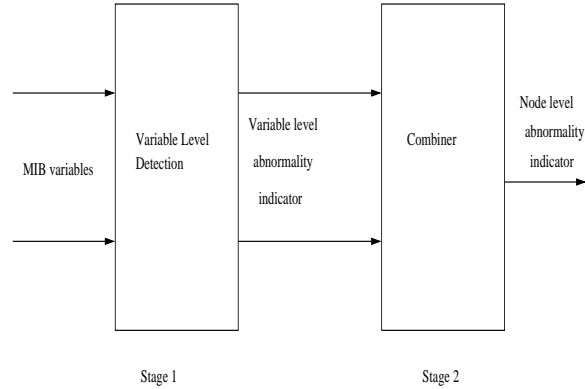


Fig. 2. Model of the Intelligent Agent

first stage detects abnormal changes at the variable level using the differenced MIB data. A time-correlated indicator of the abnormality level is produced for each of the variables. These indicators, which are computed based on the Generalized Likelihood Ratio (GLR) test and scaled between  $[0,1]$ , correspond to variable level probabilities of abnormality. The variable level indicators are used to construct an input vector which is fed into the second stage, called the combiner. The combiner incorporates spatial correlation from the variable level to compute a scalar indicator of abnormality for the network node. This indicator, which is also bounded between  $[0,1]$ , is interpreted as a measure of the probability of abnormality in the network node.

#### A. Choice of Variables

The Management Information Base variables (MIB II), which are standardized for the Simple Network Management Protocol (SNMP) version (1), fall into different groups. The Internet Protocol (*ip*) group variables were determined sufficient to describe the functionality of the router [17][18].

The variables used in the intelligent agent represent cross sections of the traffic at different points in the *ip* layer. The variables *ipIR* (In Receives) represents the total number of datagrams received from all interfaces of the router, *ipIDe* (In Delivers) represents the number of datagrams correctly delivered to the higher layers, as this node was their final destination, and *ipOR* (Out Requests) represents the number of datagrams passed on from the higher layers of the node to be forwarded by the *ip* layer. The MIB variables chosen, although non-redundant, are not strictly independent and the relationships between them have been incorporated at the combination stage described in Section IV.

### III. STAGE 1: TEMPORALLY CORRELATED VARIABLE LEVEL DETECTION

The increments in the MIB variable data constitutes a time series. The time series data for each variable was processed independently using a sequential change detection algorithm [2]. The underlying premise is that the statistical properties of the MIB variables change in response to impending fault conditions [10][7][15]. Since these changes are subtle they cannot be captured by conventional adaptive thresholding schemes that use only the mean and variance of the raw data [10]. Figure 3 shows a typical data trace of a MIB variable during normal functioning of the network and during a fault period (the asterisks denote the fault period as identified by *syslog* messages). At first glance, the two data series seem indistinguishable. Our challenge was to detect the sub-

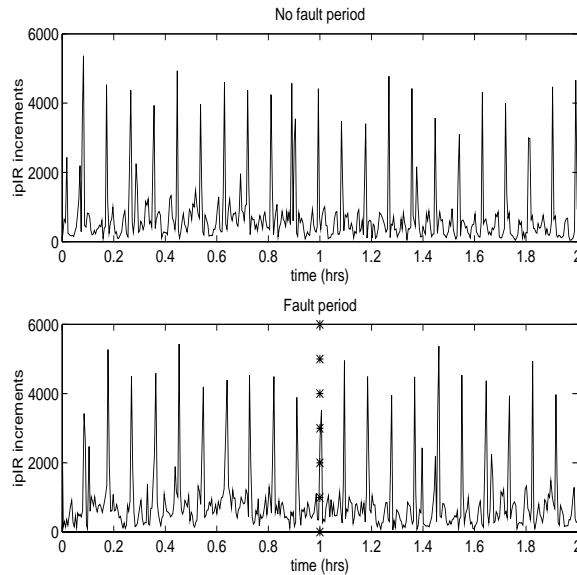


Fig. 3. Representative trace of *ipIR* variable

tle changes that precede the fault in the presence of intrinsically non-stationary behavior in the variables. To achieve this, the Auto-Regressive (AR) model was used. The AR parameters go beyond the mean and variance by including the dependency structure in the underlying time series over a short range.

Piecewise stationary AR models are commonly used to describe non-stationary stochastic time series signals [4]. The input MIB data were sequentially processed by considering the time series of each of the variables over piecewise stationary windows. Within a given window the MIB data were linearly modelled using a first order AR process. Using two adjacent piecewise stationary windows, the learning window  $L(t)$  and the test window  $T(t)$ , a sequential hypothesis test was performed using the Generalized Likelihood Ratio (GLR) test [1] [6]. The complete derivation of the test statistic can be found in [17] [18].

The joint likelihood  $l$  of the residual errors in the two windows  $L(t)$  and  $T(t)$  of length  $N_L$  and  $N_T$  respectively is given as,

$$l = \left( \frac{1}{\sqrt{2\pi\sigma_L^2}} \right)^{\dot{N}_L} \left( \frac{1}{\sqrt{2\pi\sigma_T^2}} \right)^{\dot{N}_T} \exp \left( \frac{-\dot{N}_L \hat{\sigma}_L^2}{2\sigma_L^2} \right) \exp \left( \frac{-\dot{N}_T \hat{\sigma}_T^2}{2\sigma_T^2} \right), \quad (1)$$

where  $\sigma_L^2$  and  $\sigma_T^2$  are the variance of the residuals in windows  $L(t)$  and  $T(t)$ ,  $\dot{N}_L = N_L - p$ ,  $\dot{N}_T = N_T - p$  and  $\hat{\sigma}_L^2$  and  $\hat{\sigma}_T^2$  are the covariance estimates of  $\sigma_L^2$  and  $\sigma_T^2$  [6]. The expression for  $l$  is a sufficient statistic and is used to perform a binary hypothesis test. Under the hypothesis  $H_0$ , implying that no change is observed between the two windows, we have the likelihood  $l_0$ :

$$l_0 = \left( \frac{1}{\sqrt{2\pi\sigma_P^2}} \right)^{\dot{N}_L + \dot{N}_T} \exp \left( \frac{-\left(\dot{N}_L + \dot{N}_T\right) \hat{\sigma}_P^2}{2\sigma_P^2} \right) \quad (2)$$

where  $\sigma_P^2$  is the pooled variance. Under hypothesis  $H_1$ , implying that a change is observed between the two windows we have,  $l_1 = l$ . In order to obtain a value for the likelihood ratio  $\eta$  that is bounded between [0 1], we define  $\eta$  as follows,

$$\eta = \frac{l_1}{l_1 + l_0} \quad (3)$$

Furthermore, on using the maximum likelihood estimates for the variance terms in equations (1) and (2) we get;

$$\eta = \frac{\hat{\sigma}_L^{-\dot{N}_L} \hat{\sigma}_T^{-\dot{N}_T}}{\hat{\sigma}_L^{-\dot{N}_L} \hat{\sigma}_T^{-\dot{N}_T} + \hat{\sigma}_P^{-(\dot{N}_L + \dot{N}_T)}} \quad (4)$$

Using this approach, we obtain a sequential measure of abnormality for each of the MIB variables as the output of the first stage. These indicators, which are functions of system time, are updated every  $N_T$  lags.

The implementation of stage (1) depends on the choice of the test window size  $N_T$ , and the order of the AR process  $p$ . A trade off study on these issues was done in [18]. A study on the statistical properties of the residuals of the adjacent windows can be found in [17]. The length of the learning window  $N_L$  was experimentally optimised for the MIB variables, *ipIDe*, and *ipOR* to be 120 mins. The variable *ipIR* had an optimal learning window of 5 mins. We believe that this difference can be attributed to the bursty behavior of the *ipIR* variable.

#### IV. STAGE 2: SPATIAL CORRELATION USING A COMBINER

The goal of the combiner is to incorporate the spatial dependencies into the time correlated variable level indicators in order to compute a single scalar value that is predictive and represents the probability of node level abnormality. In most alarm correlation

and fault identification schemes [12][16] some type of fault model is required to incorporate spatial dependencies. When predicting a fault, no models exist that capture the MIB variable behavior before the fault occurs. In this work we attempt to provide a combining scheme that is independent of specific fault descriptions and amenable for online implementation.

Our combining scheme consisted of an operator matrix to incorporate the spatial dependencies. In analogy to quantum mechanics [5] the observable of this operator was interpreted as the abnormality of the network and the expectation of the observable was the scalar quantity  $\lambda$  used to indicate the abnormality of the network node.

First a  $(1 \times n)$  input vector  $\psi$  was constructed with components:

$$\psi = [ \eta_1 \quad . \quad . \quad . \quad \eta_n ] \quad (5)$$

Each component of this vector corresponds to the probability of abnormality associated with each of the MIB variables. In order to complete the basis set so that all possible states of the system are included, an additional component  $\eta_0$  that corresponds to the probability of normal functioning of the network was created. The final component allows for proper normalisation of the input vector. The new input  $\psi$  vector,

$$\psi = \alpha [ \eta_1 \quad . \quad . \quad . \quad \eta_n \quad \eta_0 ] \quad (6)$$

was normalised with  $\alpha$  as the normalisation constant. By normalising the input vector, we obtain a value between [0 1] for the expectation of the observable  $\lambda$ , which we interpreted as the probability of node level abnormality.

The operator matrix  $A$  was designed to be Hermetian. The entries of the matrix show how the operator causes the components of the input vector to interact with each other. Since matrix  $A$  is Hermetian, its eigenvectors  $\phi_i$  are orthogonal. Once normalized, these eigenvectors were used to form an orthonormal basis set. Therefore any input vector  $\psi$  can be decomposed onto its eigenvector basis as follows:

$$\psi^T = \sum_{i=1}^n c_i \phi_i \quad (7)$$

The input vector  $\psi^T$  that is transformed by the operator  $A$  can be written as

$$A\psi^T = A \sum_{i=1}^n c_i \phi_i \quad (8)$$

$$= \sum_{i=1}^n c_i \lambda_i \phi_i \quad (9)$$

where  $\lambda_i$ , are the eigenvalues of  $A$ . In order to obtain a scalar value of the measure of the transformation we perform the following operation:

$$\psi A \psi^T = \sum_{i=1}^n c_i^2 \lambda_i \quad (10)$$

$$= E(\lambda) \quad (11)$$

where  $c_i$  is the amplitude of the projection of any input vector  $\psi$  onto the  $i$ -th eigenvector. This quantity  $\psi A \psi^T$  provides the scalar value that corresponds to the expectation of the eigenvalue  $E(\lambda)$ .

#### A. Design and Interpretation of The Operator Matrix

At the router three variables (viz)  $ipIR$ ,  $ipIDe$ , and  $ipOR$  were considered. Including the normal probability, a  $1 \times 4$  input vector was required:

$$\psi_{ip} = \alpha_R \begin{bmatrix} \eta_{IR} & \eta_{IDe} & \eta_{OR} & \eta_{ipnormal} \end{bmatrix}. \quad (12)$$

The input vector corresponding to a completely faulty probability is  $\psi = \alpha_R \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}$  (the fourth component is 0, since the system is completely faulty). Using this vector the normalization constant  $\alpha_R$  for the router was calculated to be  $\frac{1}{\sqrt{3}}$ .

The appropriate operator matrix  $A_{ip}$  will be  $4 \times 4$ . We design the operator matrix to be Hermetian. Taking the normal state to be uncoupled to abnormal states we get a block diagonal matrix with a  $3 \times 3$  upper block  $A_{ip_{upper}}$  and a  $1 \times 1$  lower block:

$$A_{ip} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

The  $a_{44}$  element indicates the contribution of the healthy state to the indicator of abnormality for the network node ( $E[\lambda]$ ). Since the healthy state should not contribute to the abnormality indicator, we assigned  $a_{44} = 0$ .

The elements in the upper block  $A_{ip_{upper}}$  represent the interaction between the abnormal states of the MIB variables under the action of the operator (the element  $a_{mn}$  is the projection of  $A_{ip_{upper}} \psi_n$  onto the basis vector  $\psi_m$ ). The elements  $a_{mn}$  of  $A_{ip_{upper}}$  were assigned based on the spatial correlation between the variables. The coupling of the  $ipIR$  variable with  $ipOR$  and  $ipIDe$  variables ( $a_{12}$  and  $a_{13}$ ) were assigned values 0.08 and 0.05 respectively. This was because the majority of packets received by the router are forwarded at the  $ip$  layer and not sent to the higher layers. The coupling between  $ipIDe$  and  $ipOR$  ( $a_{23}$ ) is significantly higher since both variables relate to router processing which is performed at the higher layer. These assignments were based on the flow of traffic and the statistical correlations between the variables. By symmetry:  $a_{21} = a_{12}$ ,  $a_{31} = a_{13}$ , and  $a_{23} = a_{32}$ . The main diagonal terms are assigned such that the rows and columns sum to 1. Thus our  $A_{ip_{upper}}$  matrix becomes:

$$A_{ip_{upper}} = \begin{bmatrix} 0.87 & 0.08 & 0.05 \\ 0.08 & 0.6 & 0.32 \\ 0.05 & 0.32 & 0.63 \end{bmatrix}$$

Note that the lower block does not affect the indicator of network abnormality. Hence our computation only uses the upper block. Therefore equation(11) becomes:

$$E[\lambda] = \psi_{upper} A_{ip_{upper}} \psi_{upper}^T \quad (13)$$

where  $\psi_{upper} = \alpha_R \begin{bmatrix} \eta_{IR} & \eta_{IDe} & \eta_{OR} \end{bmatrix}$ .

#### Geometric Interpretation

The eigenvalues of the upper block matrix  $A_{upper}$  are  $\lambda_1 = 0.2937$ ,  $\lambda_2 = 0.8063$ , and  $\lambda_3 = 1$ . The corresponding eigenvectors are  $\phi_1 = [-0.0414 \ 0.7269 \ -0.6855]$ ,  $\phi_2 = [0.8154 \ -0.3718 \ -0.4436]$ , and  $\phi_3 = [0.5774 \ 0.5774 \ 0.5774]$ . These vectors are shown in Figure 4. The fourth eigenvector, which is not shown is  $\phi_4 = [0 \ 0 \ 0 \ 1]$  with eigenvalue  $\lambda_4 = 0$ . The cube shown in the first sector of

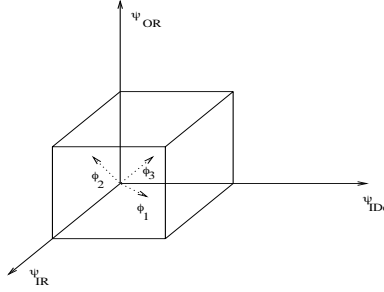


Fig. 4. Eigenvectors and Problem Domain

the three dimensional space in Figure 4 represents the problem domain. This is because the input variables to the combiner range from  $[0 \ 1]$ . The eigenvector  $\phi_3$  corresponds to the total fault vector ( all input components abnormal) and is present at the center of the cube. Eigenvectors  $\phi_1$  and  $\phi_2$  are necessarily outside the problem domain since they must be orthogonal to  $\phi_3$ . Thus in our problem, unlike in Quantum Mechanics, two of the eigenvectors are outside the problem domain: however projections of  $\psi$  onto  $\phi_1$  and  $\phi_2$  are allowed.

Suppose the input vectors were only composed of  $\phi_2$  and  $\phi_3$ , then

$$\psi = c_2\phi_2 + c_3\phi_3 \quad (14)$$

Since  $\psi$  was normalised,

$$c_2^2 + c_3^2 = 1 \quad (15)$$

Substituting  $\psi$  into Equation ( 11) we get the abnormality indicator:

$$E[\lambda] = c_2^2\lambda_2 + c_3^2\lambda_3. \quad (16)$$

In this case  $E[\lambda]$  is bounded by  $\lambda_2 = 0.8063$  and  $\lambda_3 = 1$ . This result led us to use  $\lambda_2$  as the threshold to indicate node level alarms. Note that input vectors which are not composed exclusively by  $\phi_2$  and  $\phi_3$  could still yield an  $E[\lambda] > \lambda_2$ , but these vectors would necessarily have large projections on  $\phi_2$  and/or  $\phi_3$ . The abnormal region is defined as:

$$\lambda_2 < E[\lambda] \leq \lambda_3 \Rightarrow \text{abnormal region} \quad (17)$$

## V. EXPERIMENTAL WORK

The experiments were conducted on the Local Area Network (LAN) of the Computer Science (CS) Department at Rensselaer Polytechnic Institute. The network topology is

as shown in Figure 5. The CS network forms one subnet of the main campus network. Within the CS network there are seven smaller subnets and two routers. One router is used for internal routing and the other serves mainly as a gateway to the campus backbone. The internal router has 6 interfaces with the CS subnets. The majority of the

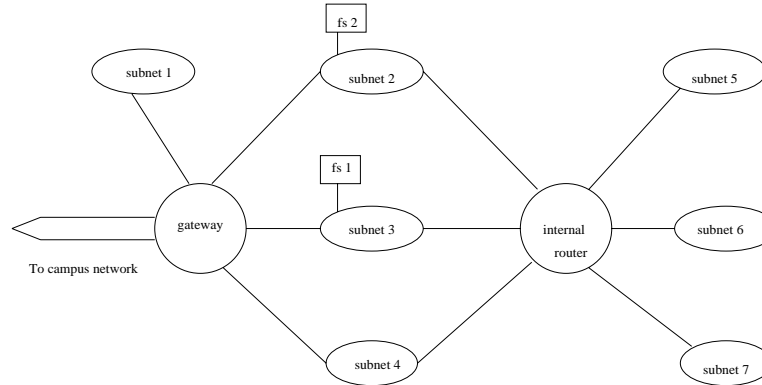


Fig. 5. Configuration of the monitored network

traffic is file transfers or web traffic which involves the workstations accessing the file servers. The internal router is an SNMP agent. The Management Information Base on this agent was polled (using a PERL script) every 15 seconds to obtain the measurement variables. Data was collected from the internal router.

There were no network management measures in place on this network. Each machine on the network ran the UNIX *syslog* function. This function generated messages that related to problems associated with the network, applications, or the specific machine itself. These *syslog* messages were used to identify the network problems. One of the most common network problems was NFS server not responding. The *syslog* messages only reported that the file server was not responding, but was unable to identify the cause of the problem. Possible reasons for this problem are unavailability of network path or that the server was down. Although not all problems could be associated with *syslog* messages, those problems which were identified by *syslog* messages were accurately correlated with fault incidents. A description of the data sets used is provided in the Table( I). In most cases the agent was able to predict the occurrence of a fault significantly ahead of the *syslog* message reports.

#### A. Case Study (1):

Here we describe a fault scenario corresponding to a file server failure on subnet 2 (data set 2 fault 1 in Table I). 12 machines on subnet 2 and 24 machines outside subnet 2 reported the problem via *syslog* messages. The duration of the fault was from 11.10am to 11.17am (7mins) as determined by the *syslog* messages. The cause of the fault was confirmed to be excessive number of *ftp* requests to the specific file server. Figure 6 shows the output of the agent at the router and at the *ip* layer variable level.

TABLE I  
DESCRIPTION OF FAULT DATA SETS: FAULT LOCATION AND TIME

Data set no	No of faults in data set	Fault location (subnet)	Time and/ duration of faults	Number of machines outside fault subnet
1	1	2	4.19 - 4.21pm	24
2	1	2	11.10 - 11.17am	24
3	3	2	8.23 - 8.26pm	14
		2	1.23 - 1.25am	0
		2	9.48 - 9.52am	15
4	1	2	6.33 - 6.36am	6
5	1	2	9.36 - 9.41pm	10
6	2	3	11.17 - 11.21pm	2
		3	11.28 - 11.30pm	
		3	3.22 - 3.26pm	3

The indicators provide the trends in abnormality. The variable level abnormality indicator contains the temporal information. The fault period is shown by the vertical dotted lines. In Figure 6 for router health, the 'x' denotes the alarms that correspond to input vectors that are faulty. Note that there are very few such alarms at the router level. The mean time between false alarms in this case was found to be 98 mins. The fault was predicted 29 mins before the crash occurred. The persistence in the abnormal behavior of the router is also captured by the indicator. The on-off nature of the *ipIDE* and *ipOR* indicators was attributed to the less bursty behavior of those variables. Note also that the router shows abnormal behavior soon after the fault. This was attributed to the hysteresis. In our present scheme no measures are taken to combat this effect.

#### B. Case Study (2):

This case corresponds to a file server failure on subnet 3 (data set 6 fault 2 in Table I). 8 machines on subnet 3 and 3 machines outside subnet 3 reported the problem. The duration of the fault was from 3.22pm to 3.26am (4 mins). Figure 7 shows the output of the agent. In this case the mean time between false alarms was found to be 34 mins. The fault was predicted less than 1 minute before the crash occurred. However the persistence in abnormal behavior of the router was observed several minutes before the actual crash.

In the above two cases we have shown that the agent is capable of predicting faults at different times of the day. Regardless of the number of machines that are affected (24 in the first case and 3 in the second case) outside the subnet, the agent is able to predict the problem as long as there is some traffic that affects the network layer variables.

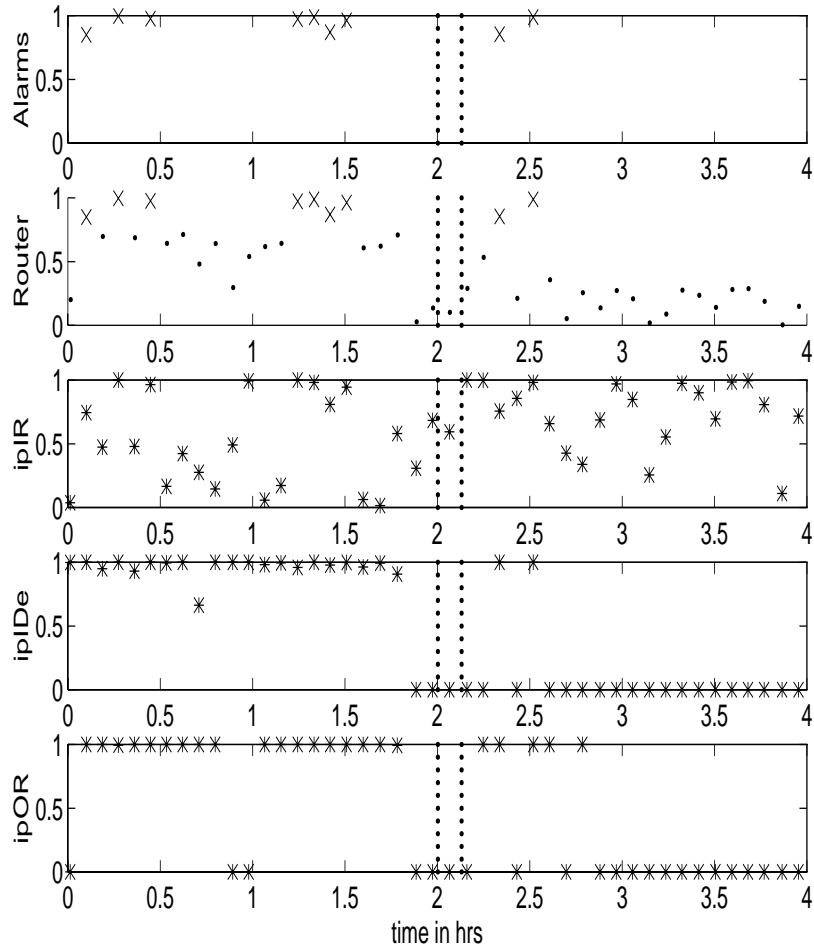


Fig. 6. Abnormality indicators at the Router

## VI. PERFORMANCE OF THE AGENT AND COMPOSITE RESULTS

The performance of the algorithm is expressed in terms of the probability of prediction  $P_p$ , prediction time  $T_p$ , and the mean time between false alarms  $T_f$ .

$$P_p = \frac{\text{Total Number of True Alarms}}{\text{Total Number of Known Faults}} \quad (18)$$

We distinguished between a true alarm and a false alarm as follows: a true alarm corresponds to a set of one or more consecutive alarms subject to the following constraints:

$$\tau < 15\text{mins} \quad (19)$$

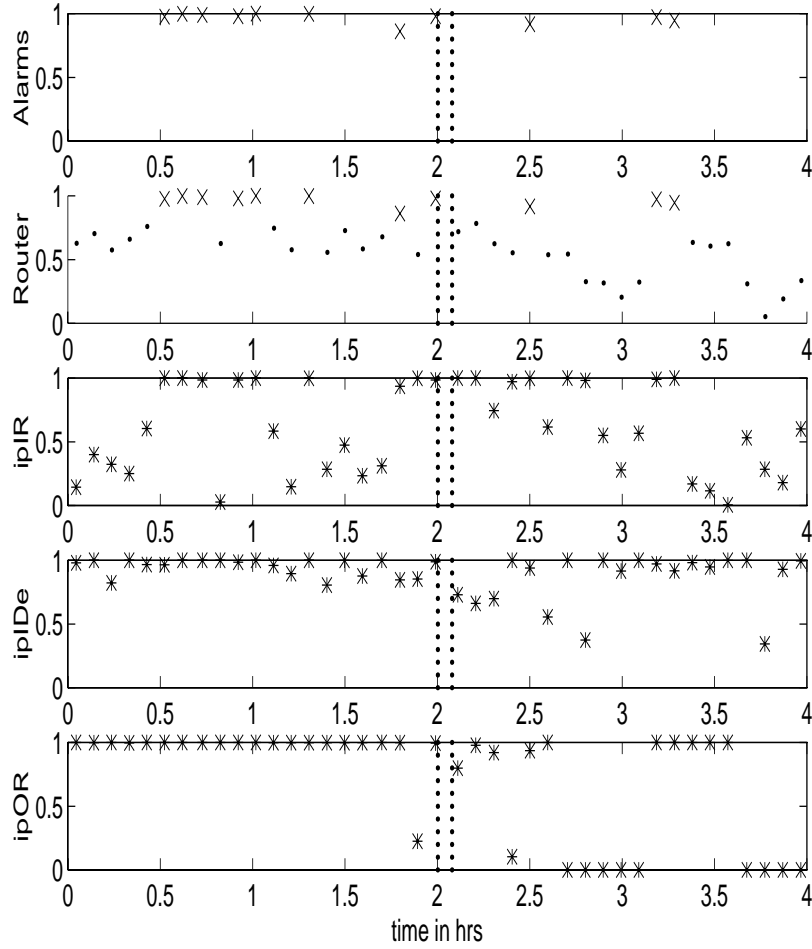


Fig. 7. Abnormality indicators at the Router

$$T_p < T_f \quad (20)$$

where  $\tau$ , is the time between any two consecutive alarms and corresponds to 3 lags. These quantities are depicted in Figure 8. The composite results for the data obtained from the internal router are compiled in Table (II). Note the average prediction time (23 mins) is less than half the mean time between false alarms (52 mins). The time scale of prediction is large enough to allow time for potential corrective measures. Seven out of nine faults were predicted making the average probability of prediction  $P_p = 0.78$ . In data set 3, fault 2 was reported by only two machines on the same subnet on which the faulty file server was located. This suggests that for this fault there was minimal impact on the *ip* level traffic which resulted in no prediction.

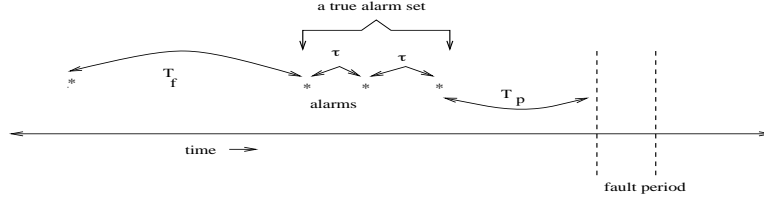


Fig. 8. Quantities Used in Performance Analysis

Data Set no	Fault No	Prediction Time $T_p$ (mins)	Mean Time Between False Alarms $T_f$ (mins)	$P_p$
1	1	-	53	0
2	1	29	98	1
3	1	34	55	0.67
	2	-		
	3	16		
4	1	29	37	1
5	1	27	32	1
6	1	23	34	1
	2	<1		
Avg		23	52	0.78

TABLE II  
PREDICTION OF FAULTS AT THE INTERNAL ROUTER

The algorithm is based on a linear model, rendering it feasible for online implementation. The complexity as a function of the number of model parameters is  $O(4N)$ , where  $N$  is the number of input MIB variables. For the work discussed here we have  $N = 3$ . The computational complexity expressed in terms of the number of floating point operations performed is approximately 9 per sec.

## VII. DISCUSSION AND CONCLUSION

The intelligent agent was shown to be capable of predicting network problems with a very high probability. The agent captures most of the salient features of fault behavior. The time scale of prediction is sufficient to allow for triggering corrective mechanisms to alleviate the impending problem. The agent has been implemented in an online fashion.

The current architecture allows the agent to scale easily to multiple nodes. The variable level detection was done using sequential testing. The variable level indicators were designed to maximize information by avoiding thresholds. Since strict modelling of the signal was not required but rather accurate prediction and simplicity, an AR(1) model was used. The new combination scheme incorporates the spatial dependencies more naturally than the duration filter used previously [17] and as expected performed

better under the stricter criteria for prediction used in this paper. The idea of fault vectors provides a structure for understanding the multivariable input vectors as well as a good criteria for declaring node level alarms.

Future efforts will be made in adapting the agent to gateways and switches. A similar agent has already been developed for the interface level traffic. Using information from both the interface *if* and the network *ip* layers we were able to isolate the problem to the subnet level and locate the possible origin of the problem. Work is under way to combine the information from the *ip* and the *if* layers to reduce false alarms and obtain the average abnormality of the entire network. Efforts will also be concentrated on finding better time series models for the bursty variable(*ipIR*). We are also working on fault simulations to aid the bench marking of the agent. Currently data obtained from a larger enterprise network is being used to test the scalability of the agent.

#### VIII. ACKNOWLEDGMENTS

The authors would like to thank Dave Hollinger, Nathan Schimke, Roddy Collins and Cindy Hood for their generous help with the data collection. We also thank Ashok Malikal and Ken Vastola for helpful discussions on the topic. The support of the National Science Foundation ((CAREER) IRI-9502518) and DARPA (F30602-97-C-0274) is gratefully acknowledged.

#### REFERENCES

- [1] U. Appel and A.V. Brandt. Adaptive sequential segmentation of piecewise stationary time series. *Information Sciences*, 29:27–56, 1983.
- [2] M. Basseville and A. Benveniste. Sequential segmentation of non stationary digital signals using spectral analysis. *Information Sciences*, 29:57–73, 1983.
- [3] A. Bouloutas, G. Hart, and M. Schwartz. *On the Design of Observers for fault Detection in Communication Networks, Network Management and Control*. Plenum Press, New York, 1990.
- [4] Box and Jenkins. *Time Series Analysis, Forecasting and Control*. Holden Day Series, 1976.
- [5] C. Cohen-Tannoudji, B. Diu, and F. Laloe. *Quantum Mechanics*, volume 1. A Wiley Interscience Publication, 2 edition, 1977.
- [6] P.V. Desouza. Statistical tests and distance measures for lpc coefficients. *IEEE trans on Acoustics, Speech and Signal Processing*, 25(6), Dec 1977.
- [7] F. Feather and R. Maxion. Fault detection in an ethernet network using anomaly signature matching. In *ACM SIGCOMM*, volume 23, Sept 1993.
- [8] G. Goldszmidt and Y. Yemini. Distributed management by delegation. In *15th International Conference on Distributed Computing*, 1995.
- [9] C. Hood and C. Ji. Automated proactive anomaly detection. In *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, pages 688–700, 1997.
- [10] C.S. Hood and C. Ji. Proactive network fault detection. *Proceedings of INFOCOM, Kobe, Japan*, 1997. Also available from <http://neuron.ecse.rpi.edu/>.
- [11] I. Katzela. *Fault Diagnosis in Telecommunication Networks, Doctoral Thesis*. Columbia University, New York, N.Y. USA, 1996.
- [12] I. Katzela and M. Schwarz. Schemes for fault identification in communication networks. *IEEE/ACM Trans.Networking*, 3:753–764, 1995.
- [13] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. In *ACM SIGCOMM*, 1997.
- [14] A. Lazar, W. Wang, and R. Deng. Models and algorithms for network fault detection and identification: A review. In *Proc. IEEE ICC*, 1992.
- [15] R. Maxion. A case study of ethernet anomalies in a distributed computing environment. *IEEE transactions on Reliability*, 39(4), Oct 1990.
- [16] I. Rouvellou and G.W. Hart. Automatic alarm correlation for fault identification. In *Proc. IEEE INFOCOM*, pages 553–561, 1995.

- [17] M. Thottan and C. Ji. Adaptive thresholding for proactive network problem detection. In *Proceedings of IEEE International Workshop on Systems Management, Newport, Rhode Island*, 1998. Also available from <http://neuron.ecse.rpi.edu/>.
- [18] M. Thottan and C. Ji. Statistical detection of enterprise network problems. *Journal of Network and Systems Management, in press*, 1999. Also available from <http://neuron.ecse.rpi.edu/>.

#### IX. BIOGRAPHY

Marina Thottan received her M.S. in Physics at P.S.G. College of Arts and Science, Coimbatore, India. She received her M.S. in Biomedical Engineering at Rensselaer Polytechnic Institute where she is currently working towards a Ph.D in Electrical and Computer Systems engineering.  
(<http://www.rpi.edu/~thottm>)

Chuanyi Ji received the BS (with honors) from Tsinghua University, Beijing, China in 1983, an MS from University of Pennsylvania, Philadelphia in 1986, and a Ph.D from California Institute of Technology, Pasadena, in 1992, all in Electrical Engineering. In November 1991, she joined the faculty at Rensselaer Polytechnic Institute, Troy, where she is now an Associate Professor of Electrical, Computer and Systems Engineering. Her research interests are in the areas of stochastic modeling, analysis and control of multi-media traffic, management of hybrid computer communication networks, and adaptive learning systems. Dr. Ji is a recipient of NSF CAREER award in 1995.

(<http://www.ecse.rpi.edu/homepages/chuanyi>)