

Using Network Fault Predictions to Enable IP Traffic Management*

M. Thottan

Lucent Technologies, Bell Laboratories,
Department of Network and Service Management
Holmdel, NJ 07733

C. Ji

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

April 20, 2001

Abstract

IP traffic management is important for the continued growth of the Internet. Today there exist several traffic management algorithms. However, in order to enable these algorithms it is necessary to provide reliable alarms relating to network performance bottle necks and failures. In this work we propose an algorithm to obtain reliable predictive alarms for network fault conditions. The algorithm is based on modeling network fault behavior. The algorithm has been successfully tested on two production networks. Predictive alarms were obtained for four different types of failures: file server failures, network access problems, protocol implementation errors and runaway processes. The potential of using this model to do fault classification is also discussed. In addition, we show that the proposed algorithm performs better than the majority-vote scheme.

Running Header: Fault Predictions to Enable Traffic Management

Keywords: network health, change detection, fault model, spatial correlator

1 Introduction

Efficient management of network traffic is important for the continued growth of the Internet. Research in the area of IP traffic management has primarily focussed on developing management algorithms for specific tasks such as admission control [7], policing and shaping of network traffic [20]. These algorithms

*Supported by DARPA under contract number F30602-97-C-0274

can be deployed as soon as an alarm is obtained from the system notifying the existence of a performance bottle-neck or a network fault. However the problem of generating these alarms has received little attention. This paper addresses the issue of alarm generation by providing a method of obtaining an online notification of the health of the network. The health indicator is obtained by modeling network fault behavior.

One of the difficulties encountered in automating IP traffic management is the lack of confidence in network alarms. Therefore alarms that are used to trigger a management decision must have a very high confidence level. This is an essential requirement in order to maintain the stability of the network. The non-stationary behavior of network traffic makes it difficult to attain high confidence levels in the alarms generated. Currently there is no method for generating alarms that are capable of identifying several different types of network problems. In this work we propose an algorithm that generates an indicator of network health. The health indicator obtained is more efficient than majority vote schemes and also has the potential to recognize several types of failures and performance problems.

In order to generate alarms for triggering IP traffic management algorithms, we measure traffic variables to detect failure conditions and performance bottle-necks. Due to the evolving nature of network devices and traffic statistics the modeling of the normal behavior of network traffic is non-trivial. Therefore, in our work we chose to model network fault behavior. We hypothesize that network failures are preceded by identifiable changes in network traffic metrics. Since these changes occur prior to the fault they are predictable. Furthermore these failures cause persistent correlated abrupt changes to the traffic related variables in the system. In this paper we show experimental evidence for the hypothesis and also propose an algorithm that uses this model to provide a continuous indicator of network health. This indicator provides alarms that can be used to trigger online IP traffic management strategies. In addition the indicators are capable of classifying the fault behavior. The algorithm described in this paper was designed to capture any traffic related anomalies. It is independent of specific fault descriptions and therefore has the ability to detect a wide variety of traffic related faults.

The paper is organized as follows: Section 2 discusses the fault model and shows experimental evidence of the hypothesis and a mathematical representation of the same. The proposed algorithm to obtain continuous network health indication is discussed in section 3. The experimental results obtained using data from two different production networks is presented in section 4. This is followed by section 5 where we compare the health indicator to the standard majority vote scheme. We also show the potential for using the indicator for fault classification purposes.

2 Fault Model

One of the main challenges faced by the research community is the choice of a single variable or a set of variables that are relevant towards fault detection. Existing management tools provide statistics on a large number of variables that may or may not be relevant to fault detection. Data mining techniques are being used to study different management databases in order to extract the relevant information [8]. Statistical information obtained from such variables constitute the feature vectors. For any given fault detection scheme, to cover a wide range of failures it is necessary to choose a set of relevant feature vectors. In the case of an Ethernet, Maxion et al used features such as packet loss, and number of collisions [10]. In [6], for a transaction- oriented network a 3-tuple feature set consisting of the start time and duration of the transaction along with the service class identifier was used. Trouble tickets can be used as feature vectors in algorithms using artificial intelligence techniques [9],[13], [12]. The focus of our work is on traffic related problems. Hence MIB data (Management Information Base variables [11], which are a part of SNMP) and closely correspond to network traffic are used. For example in the case of the router we used the following MIB variables;

ipIR: Number of datagrams received by the *ip* layer of the router

ipIDe: Number of datagrams forwarded to the higher layers

ipOR: Number of datagrams received from the higher layers.

These MIB variables can be discovered by using simple principal component analysis techniques [5]. Intuitively, these variables provide a cross-sectional view of the traffic at the network layer. For more details on the choice of these variables refer [15].

2.1 Hypothesis

Our hypothesis is that, *network performance problems and failures are preceded by persistent and correlated abrupt changes in the traffic related MIB variables*. A schematic of this hypothesis is shown in Figure 7. Abrupt changes refer to significant changes in the MIB variable statistics at a time scale comparable to the sampling rate. In this case the sampling rate is 15 secs. Hence abrupt changes are changes that occur in the order of mins. These abrupt changes occur spatially correlated across the different MIB variables. They also persist for some time prior to the occurrence of the fault.

2.2 Experimental Evidence

To better motivate our model of network faults we will discuss in detail a specific fault manifestation. This particular fault occurred on a campus LAN network and corresponded to a file server failure that was reported by 36 machines of which 12 were located on the same subnet as the file server. The fault lasted for seven minutes. Figures 7 through 7 show the trace of the different traffic-related MIB variables at the *ip* layer, 2 hours before the fault was observed by

Figure 1: Schematic for Fault model

existing mechanisms such as *syslog* messages. This particular fault is a good illustrative case as the deviations from normal network behavior are more easily observable in the traffic traces. The extent of deviation from normal behavior is different for different variables and also varies based on the manifestation of the fault. The changes observed in the *ipIR* variable are much more subtle than the changes in the *ipIDe* and *ipOR* variables.

Another important aspect to be noted is that the subtle abrupt changes associated with the fault events occur spatially correlated across the different MIB variables. Note in Figures 7 through 7 that there are abrupt changes observed in all the three *ip* level variables less than one half hour before the fault occurred.

2.3 Mathematical Model

The abrupt changes in the MIB variables can be modeled using an Auto-Regressive (AR) process [3]. These abrupt changes occur spatially correlated among the different MIB variables and persist over a certain time lag τ . This persistent correlation property distinguishes abrupt changes intrinsic to fault situations from those random changes of the system which are related to the network's normal function. Therefore network fault events occur at some time t_a such that:

$$t_a = \inf\left\{t : \sum_{j=0}^{\tau} I_{\lambda_{f_{min}} \leq f_j(\vec{\psi}(t)) \leq \lambda_{f_{max}}} \geq \beta\right\} \quad (1)$$

inf here refers to the minimum value of t that satisfies the condition specified in equation(1). The abrupt changes in the characteristics of the MIB variables are captured by the Generalized Likelihood Ratio (GLR) test [19]. The ratio pro-

Figure 2: Trace of $ipIR$ Before Fault. Fault Period Denoted by Asterisks. Bold Line Denotes Changes in Time Series Preceding the Fault

Figure 3: Trace of $ipIDe$ Before Fault. Fault Period Denoted by Asterisks. Bold Line Denotes Changes in Time Series Preceding the Fault

Figure 4: Trace of $ipOR$ Before Fault. Fault Period Denoted by Asterisks. Bold Line Denotes Changes in Time Series Preceding the Fault

vides an abnormality indicator that is scaled between 0 and 1. The abnormality indicators are collected to form an abnormality vector $\vec{\psi}(t)$. The abnormality vector $\vec{\psi}(t)$ is a measure of the abrupt changes in normal network behavior. The spatial dependencies of the abrupt changes between the individual MIB variables are incorporated using a quadratic functional $f_j(\vec{\psi}(t))$ that involves a linear operator A . A is a matrix which incorporates the correlation between the MIB variables. The parameters λ correspond to the eigenvalues of A . I_x is an indicator function that corresponds to network alarms and has a value of 1 when condition x is satisfied and 0 otherwise.

In particular the quadratic functional:

$$f_j(\vec{\psi}(t)) = \vec{\psi}(t)A\vec{\psi}(t), \quad (2)$$

represents a continuous scalar indicator of network health. A subset of the eigenvectors of A correspond to fault states in the network. For example the eigenvector whose components correspond to maximum abnormality values corresponds to a fault state with eigenvalue 1. This vector is taken to be the fault vector because it signifies a situation of high spatially correlated abnormality. $\lambda_{f_{min}}$ and $\lambda_{f_{max}}$ are the minimum and maximum eigenvalues that correspond to these fault states. The summation of the indicator over τ time lags captures the persistence of the correlated abrupt changes. The value of β captures the degree of persistence observed in the correlated abrupt changes.

3 Proposed Algorithm for Network Health Indication

The increments in the MIB variable data constitute a time series. Feature vectors are obtained using the statistical changes in the MIB variables. The proposed algorithm to detect network problems is shown in Figure 7. The individual blocks are discussed in the following subsections.

Figure 5: Schematic representation of the proposed algorithm

3.1 Change Detector

The time series data for each variable was processed independently using a sequential change detection algorithm [2]. The variables were treated independently due to their inherent non-stationarity. The change detection algorithm captures the subtle changes in the MIB variables that precede the fault despite the non-stationary behavior exhibited by the variables. To achieve this, an 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.

The 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 modeled using a first order AR process. Using two adjacent piecewise stationary windows, a sequential hypothesis test was performed using the Generalized Likelihood Ratio (GLR) test [1] [4]. The complete derivation of the test statistic can be found in [17].

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,

$$l = \left(\frac{1}{\sqrt{2\pi\sigma_L^2}} \right)^{N_L} \left(\frac{1}{\sqrt{2\pi\sigma_T^2}} \right)^{N_T} \exp \left(\frac{-N_L\hat{\sigma}_L^2}{2\sigma_L^2} \right) \exp \left(\frac{-N_T\hat{\sigma}_T^2}{2\sigma_T^2} \right), \quad (3)$$

where σ_L^2 and σ_T^2 are the variance of the residuals in windows L(t) and T(t), $\hat{N}_L = N_L - p$, $\hat{N}_T = N_T - p$ and, $\hat{\sigma}_L^2$ and $\hat{\sigma}_T^2$ are the covariance estimates of σ_L^2 and σ_T^2 [4]. p is the order of the AR parameters used. For a justification on the normality assumption used in equation(3) refer [18]. 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 :

$$l_0 = \left(\frac{1}{\sqrt{2\pi\sigma_P^2}} \right)^{\hat{N}_L + \hat{N}_T} \exp \left(- \frac{(\hat{N}_L + \hat{N}_T) \hat{\sigma}_P^2}{2\sigma_P^2} \right) \quad (4)$$

where σ_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 ψ_i (for the i th MIB variable), that is bounded between [0 1], we define ψ_i as follows,

$$\psi_i = \frac{l_1}{l_1 + l_0} \quad (5)$$

Furthermore, on using the maximum likelihood estimates for the variance terms in equations (3) and (4) we get;

$$\psi_i = \frac{\hat{\sigma}_L^{-\hat{N}_L} \hat{\sigma}_T^{-\hat{N}_T}}{\hat{\sigma}_L^{-\hat{N}_L} \hat{\sigma}_T^{-\hat{N}_T} + \hat{\sigma}_P^{-(\hat{N}_L + \hat{N}_T)}} \quad (6)$$

Using this approach, we obtain a sequential measure of abnormality ψ_i for each of the MIB variables. These indicators, which are functions of system time, are updated every N_T lags. The specific implementation issues of the change detector and the trade-off between window sizes and detection times are discussed in [18]. In summary, a learning window size N_L of approximately 2 hours (480 samples under a sampling frequency of 15 secs) and a test window size N_T of approximately 5 minutes (20 samples at 15 sec sampling) was used.

3.2 Spatial Correlation Using a Combiner

The goal of the combiner is to incorporate the spatial dependencies into the variable level indicators. Our combining scheme is independent of specific fault descriptions and amenable for online implementation. It consists of an operator matrix A which is used to capture the correlation among the variables.

First an input vector $\vec{\psi}$ was constructed with components ψ_i that correspond 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 ψ_0 that corresponds to the probability of normal functioning of the network was created. The final component allows for proper normalization of the input vector. The new input $\vec{\psi}$ vector,

$$\vec{\psi} = \alpha [\psi_1 \quad . \quad . \quad . \quad \psi_n \quad \psi_0] \quad (7)$$

was normalized with α as the normalization constant. 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 $\vec{\phi}_i$ are orthogonal. Once normalized, these eigenvectors were used to form an orthonormal basis set. Therefore any input vector $\vec{\psi}$ can be decomposed onto its eigenvector basis as follows:

$$\vec{\psi}^T = \sum_{i=1}^n c_i \vec{\phi}_i \quad (8)$$

where c_i is the projection of $\vec{\psi}$ onto $\vec{\phi}_i$. The input vector $\vec{\psi}^T$ that is transformed by the operator A can be written as

$$A\vec{\psi}^T = A \sum_{i=1}^n c_i \vec{\phi}_i \quad (9)$$

$$= \sum_{i=1}^n c_i \lambda_i \vec{\phi}_i \quad (10)$$

where λ_i , are the eigenvalues of A . For an in depth discussion on the operator refer [16].

3.2.1 Design of The Operator Matrix

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

$$\vec{\psi}_{ip} = \alpha_R [\psi_{IR} \quad \psi_{IDe} \quad \psi_{OR} \quad \psi_{ip_{normal}}] . \quad (11)$$

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

The appropriate operator matrix A_{ip} will be 4×4 . Taking the normal state to be un-coupled to abnormal states we get a block diagonal matrix with a 3×3 upper block $A_{ip_{upper}}$ and a 1×1 lower block:

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

Since the healthy state should not contribute to the abnormality indicator, we assigned $a_{44} \approx 0$. The elements a_{mn} of $A_{ip_{upper}}$ were assigned based on the cross 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 low correlation value can be explained since, 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. 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 $A_{i_{p_u}pper}$ matrix becomes:

$$A_{i_{p_u}pper} = \begin{bmatrix} 0.87 & 0.08 & 0.05 \\ 0.08 & 0.6 & 0.32 \\ 0.05 & 0.32 & 0.63 \end{bmatrix}$$

3.3 Declaration of Alarms Using a Discriminant Function and Persistence Criteria

A discriminant function is used to discriminate between two classes of data: a fault class and a non-fault class. Often the discriminant function is a function of the input feature vectors and incorporates information specific to the problem being studied. A persistence filter was added in order to capture the persistent behavior of the spatially correlated changes in the MIB variables. This has the added benefit of improving the robustness of the alarms generated.

3.3.1 Discriminant Function

The discriminant function provides a scalar value of the measure of network health. The discriminant function used here was:

$$\vec{\psi}A\vec{\psi}^T = \sum_{i=1}^n c_i^2 \lambda_i \quad (12)$$

$$= E(\lambda) \quad (13)$$

where c_i is the amplitude of the projection of any input vector $\vec{\psi}$ onto the i -th eigenvector. c_i can also be interpreted as a probability amplitude based on some empirical distribution of the underlying data. This quantity $\vec{\psi}A\vec{\psi}^T$ provides the scalar value that corresponds to the expectation of the eigenvalue $E(\lambda)$. Note that the lower block does not affect the indicator of network abnormality. Hence our computation only uses the upper block. Therefore equation(13) becomes:

$$E[\lambda] = \psi_{upper}^{\rightarrow} A_{i_{p_u}pper} \psi_{upper}^{\rightarrow T} \quad (14)$$

where $\psi_{upper}^{\rightarrow} = \alpha_R [\psi_{IR} \quad \psi_{IDe} \quad \psi_{OR}]$.

Suppose the input vectors were only composed of $\vec{\phi}_2$ and $\vec{\phi}_3$, then

$$\vec{\psi} = c_2 \vec{\phi}_2 + c_3 \vec{\phi}_3 \quad (15)$$

Since $\vec{\psi}$ was normalized,

$$c_2^2 + c_3^2 = 1 \quad (16)$$

Substituting $\vec{\psi}$ into Equation (13) we get the scalar health indicator:

$$E[\lambda] = c_2^2 \lambda_2 + c_3^2 \lambda_3. \quad (17)$$

Therefore the abnormal region is defined as:

$$\lambda_2 < E[\lambda] \leq \lambda_3 \Rightarrow \text{abnormal region} \quad (18)$$

In the case of routers, the eigenvalues of the upper block matrix $A_{ip_{upper}}$ are $\lambda_1 = 0.2937$, $\lambda_2 = 0.8063$, and $\lambda_3 = 1$. The corresponding eigenvectors are $\vec{\phi}_1 = [-0.0414 \ 0.7269 \ -0.6855]$, $\vec{\phi}_2 = [0.8154 \ -0.3718 \ -0.4436]$, and $\vec{\phi}_3 = [0.5774 \ 0.5774 \ 0.5774]$. In this case $E[\lambda]$ is bounded by $\lambda_2 = 0.8063$ and $\lambda_3 = 1$. This result led us to use λ_2 as the threshold to indicate node level alarms. Note that input vectors which are not composed exclusively by $\vec{\phi}_2$ and $\vec{\phi}_3$ could still yield an $E[\lambda] > \lambda_2$, but these vectors would necessarily have large projections on ϕ_2 and, or ϕ_3 . The choice of the eigenvalue used to declare alarms involves preferential weighting of the input features based on their relevance to the nature of faults studied. In our case we are focusing on faults caused by user traffic which is maximally represented in the variable $ipIR$. Hence the eigenvalue corresponding to this variable is weighted heavier than the others and is used to declare alarms. The matrix A is designed so that the discriminant function $\vec{\psi}A\vec{\psi}^T$ returns a value between 0 and 1 [15]. The results obtained using this discriminant function are shown in Table (7).

The discriminant function described can be used to divide the problem space into fault and non-fault region. With each of the three input feature vectors ranging in value from 0 to 1, we have a problem space that is the same as a unit cube. The discriminant function carves out a region in this problem space that denotes the fault region. In general, the fault region corresponds to maximal values of abnormality in all of the feature vectors. Hence the input vector $\vec{\psi} = [111]$ corresponds to the maximum fault condition. The fault space can be represented as shown in Figure 7. The gray scale indicates the gradient in the combined abnormality of the input vectors or in network health. Thus the brighter (white) region which contains the higher values of the abnormality indicators corresponds to the highest abnormal event.

3.3.2 Persistence Criteria

The persistence in the spatially correlated abrupt changes are captured by using a persistence filter. By persistence filter we mean that given an instance of high average abnormality in the network health, an alarm is declared only when a second instance of high abnormality occurs within a specified interval of $(\tau - 1)$ lags. This persistence behavior can be taken advantage of to declare alarms corresponding to network fault situations. By incorporating persistence, we are able to significantly reduce the number of false alarms. When τ is small, all change points (normal and abnormal) are declared as an alarm but when τ is very large there are very few alarms thus, leading to a higher probability of missing a failure. For more details on the persistence criteria refer to [15].

Figure 6: Fault space (shown in lighter shade) embedded in the problem space. The axes indicate the feature vectors (abnormality indicators) obtained from the corresponding input MIB variables

4 Experimental Results Obtained Using the Discriminant Function

The experimental system consisted of two production networks: an enterprise network and a campus network. Both these networks were being actively monitored and were well designed to meet customer requirements. The types of faults observed were the following: File server failures, protocol implementation errors, network access problems and runaway processes [15]. Most of these failures were due to abnormal user activity except for the protocol implementation errors. However all of these cases did affect the normal characteristics of the MIB data, and impaired the functionality of the network.

The performance analysis of the discriminant function consisted of comparing the alarms obtained with the corresponding *syslog* messages and the trouble ticketing systems that were being actively used by the system administrators. The performance measures used were as follows: Prediction time T_p is given as,

$$T_p = T_E - T_a \tag{19}$$

where T_E is the time stamp of the fault as given by the *syslog* messages. T_a is

the alarm time given by the discriminant function. The detection time T_d is,

$$T_d = T_a - T_E \tag{20}$$

T_f , the mean time between false alarms is the average time between any two alarms obtained by the scheme that were not positively associated with a fault by the available labeling systems. The quantities T_p and T_d are constrained to be always less than T_f .

The results obtained using the discriminant function are shown in Table (7). In using the discriminant function, there is no thresholding performed on the input feature vectors prior to fault declaration. This helps to preserve the information required to detect the subtle changes associated with the different types of faults.

5 Discussion

The hypothesized network fault model was able to capture the behavior of more than one type of failure. The algorithm based on this model predicted most of the failures. The output of the algorithm is a single scalar value that has been used to generate predictive alarms corresponding to network failures. These alarms could be used to trigger the IP traffic management algorithms. The following section compares the performance of the fault model based discriminant function with a majority-vote scheme. We also show that feature vectors extracted from the MIB data can be used to classify network faults.

5.1 Comparison With Majority Voting

Majority-voting is a scheme in which alarms are declared based on a majority of the feature vectors exceeding their respective thresholds. This scheme is described in Figure 7. The scheme was implemented on data obtained from the two production networks and the results have been tabulated in Table (7). It was observed that the majority-voting scheme failed to predict or detect certain fault conditions such as network access problems, runaway processes and protocol implementation errors. On the other hand, the discriminant function predicted or detected these faults suggesting that the faults did affect the characteristics of the MIB data. The discriminant function scheme avoids hard thresholds on the input feature vectors. Therefore this scheme is able to detect the subtle changes in the MIB characteristics associated with different fault types. Imposing hard thresholding to the input feature vectors leads to a loss of information. The optimal thresholds to be imposed on the input feature vectors are hard to obtain in practice, especially with the evolving or non-stationary nature of network traffic [14]. Furthermore, the discriminant function accounts for the lesser and more subtle spatial correlation among the input feature vectors making it capable of detecting a variety of failures.

The discriminant function was able to *predict* 8 of the 9 file server failures. On the other hand, the majority-vote scheme only *detected* 8 failures at the same

--

Table 1: Prediction of failures at the router using the discriminant function scheme

--

Table 2: Detection of file server failures at the router using the majority-voting scheme

Figure 7: Majority-voting scheme for N input vectors. Σ is the sum of all thresholded feature vectors

time or immediately after it was observed by the existing mechanisms (*syslog* and trouble tickets). To provide predictability for the majority-voting scheme it will be necessary to lower the hard thresholds used. This will compromise on the number of false alarms generated. The discriminant function out-performs the majority-vote scheme by producing only half as many false alarms (the average mean time between false alarms is 8 hrs for the discriminant function and 4 hrs for the majority-vote scheme).

In addition to the benefits of prediction, the discriminant function provides a continuous indicator of network abnormality while the majority-vote scheme gives an on/off output. A continuous indicator is essential to provide trends in availability and reliability information. These trends increase the confidence in the alarms generated using the indicators. Thus on comparing Tables (7) and (7) we can conclude that a more sophisticated discriminant function that accounts for spatial correlation among the input feature vectors performs better than the majority-vote scheme.

5.2 Fault Classification

Once an alarm is obtained using the discriminant function, we sought to identify the type of impending fault. Using data obtained from the production networks, we investigated the behavior of the abnormality indicators one hour prior to the fault time. Four different faults were studied: file server failures, network access problems, protocol implementation errors and runaway process. The average values of the abnormality indicators are tabulated in Table (7). This average value is used to locate the fault in the problem space shown in Figure 7.

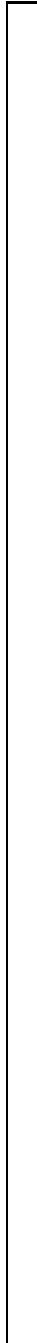


Table 3: Abnormality indicators of the feature vectors averaged over an hour prior to the fault.

Figure 8: Classification of faults using the average (over a 1 hour) of the feature vectors. x: file server failures, o: network access problems, *star*: protocol error, *square*: runaway process

As shown in Figure 7, the four fault types are clustered in different areas of the problem space. Notice that all the file server failures are clustered around the average vector $[0.4634, 0.7665, 0.8650]$. In contrast, the network access problems are clustered near $[0.5270, 0, 0]$. The Euclidean distance between these two fault clusters is approximately 1.16. The standard deviation for the network file server cluster is 0.43 and that for the network access cluster is 0.07. These results show that the two clusters do not overlap. We have limited data on the other two types of faults but it is interesting to note that they are distinct from both file server failures and network access problems. In the case of file server failures (shown as 'x') the abnormality in the *ipOR* and *ipIDe* variables are much more significant than in *ipIR*. On the contrary the network access problems (shown as 'o') are expressed only in the *ipIR* variable. The fact that these faults were predicted or detected by the discriminant function which, isolates a very narrow region of the problem space suggests that, *the abnormality in the feature vectors increases as the fault event approaches.*

6 Conclusion

We have proposed a first approach to enabling IP traffic management through modeling network fault behavior. This model has been verified both by experimental evidence and also by the successful implementation of the algorithm based on this model. The model has shown the ability to describe several different types of failures. In this work we have demonstrated using real network data, that the MIB variables show distinctive features prior to a network fault. These distinctive features can be associated with specific fault types. The four different fault types studied: file server failures, network access problems, protocol implementation errors and runaway processes, show characteristic *finger prints* in the abnormality indicators of the *ipIR*, *ipIDe* and *ipOR* MIB variables. There is sufficient distance between the clusters of file server failures and network access problems that it is possible to distinguish them easily. We believe that this is a novel approach to perform online classification of network fault conditions by looking at just an hour duration of the MIB data. It is a simple scheme and does not require much data manipulation to do classification. We only consider predictive indicators to do fault classification because we are interested in pro-actively managing the network to prevent failures. Proactive management can be done using these predictive indicators to trigger appropriate IP traffic management algorithms.

The fault classification described here can be used to develop suites of recovery options for different fault types. Furthermore, this work presents the first step to characterize network fault behavior in terms of the effects of the fault on traffic measurements. Currently research is under way to test the findings in controlled environments and on new network data. We are also in the process of linking the health indicator function to recovery measures in a test bed environment. Finally, we have shown that using discriminant functions that incorporate the spatial correlation among the MIB variables is significantly better than the majority-vote scheme.

7 Acknowledgments

The authors would like to thank Dave Hollinger, Nathan Schimke, Roddy Collins and Cindy Hood for their generous help with the campus data collection. We also acknowledge Lucent Technologies for providing data on the enterprise network. We also thank Ken Vastola for helpful discussions on the topic. The support of 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*, Vol. 29:pp 27–56, 1983.

- [2] M. Basseville and A. Benveniste. Sequential segmentation of non stationary digital signals using spectral analysis. *Information Sciences*, Vol. 29:pp 57–73, 1983.
- [3] M. Basseville and I. Nikiforov. *Detection of Abrupt Changes, Theory and Application*. Prentice Hall Information and System Sciences Series, 1993.
- [4] P.V. Desouza. Statistical tests and distance measures for lpc coefficients. *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 25, No. 6:pp 554–559, 1977.
- [5] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [6] L. L. Ho, D. J. Cavuto, S. Papavassilou, and A. G. Zawadzki. Adaptive anomaly detection in transaction-oriented wide area networks. *JNSM*, Vol. 9, No. 2:pp 761–775, June 2001.
- [7] S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang. A measurement-based admission control algorithm for integrated services packet networks (extended version). *IEEE/ACM Trans. on Networking*, Vol.5, No. 1:pp 56–70, 1997.
- [8] A. Knobbe, D. Wallen, and L. Lewis. Experiments with data mining in enterprise management. *Proceedings of IEEE/IFIP Integrated Network Management VI, Boston, USA*, pages 353–367, 1999.
- [9] L. Lewis and G. Dreo. Extending trouble ticket systems to fault diagnosis. *IEEE Network*, pages 44–51, Nov 1993.
- [10] R. Maxion and F. E. Feather. A case study of ethernet anomalies in a distributed computing environment. *IEEE Trans. on Reliability*, Vol. 39, No. 4:pp 433–443, 1990.
- [11] K. McCloghrie and M. Rose. Management information base for network management of tcp/ip-based internets: Mib 2. *RFC1213*, 1991.
- [12] C. Melchioris and L. M. R. Tarouco. Troubleshooting network faults using past experience. *Proceedings of IEEE/IFIP Network Operations and Management Symposium, Honolulu, Hawaii*, pages 549–563, 2000.
- [13] G. Penido and C. Machado J. M. Nogueira. An automatic fault diagnosis and correction system for telecommunications management. *Proceedings of IEEE/IFIP Integrated Network Management VI, Boston, USA*, pages 777–793, 1999.
- [14] D. Shen and J. Hellerstein. Predictive models for proactive network management: Application to a production web server. *Proceedings of IEEE/IFIP Network Operations and Management Symposium, Honolulu, Hawaii*, pages 833–847, 2000.

- [15] M. Thottan. *Fault Detection and Prediction for Management of Computer Networks*. Doctoral Thesis, Rensselaer Polytechnic Institute, Troy, NY, USA., May 2000.
- [16] M. Thottan and C. Ji. Fault prediction at the network layer using intelligent agents. In *IEEE/IFIP, Integrated Network Management VI, Boston, USA.*, pages 745–760, May 1999.
- [17] M. Thottan and C. Ji. Statistical detection of enterprise network problems. *Journal of Network and Systems Management*, Vol. 7, No. 1:pp 27–45, 1999. Also available from <http://neuron.ecse.rpi.edu/>.
- [18] M. Thottan and C. Ji. Adaptive thresholding for proactive network problem detection. In *Proceedings of IEEE International Workshop on Systems Management, Newport, Rhode Island*, pages 108–116, April 1998. Also available from <http://neuron.ecse.rpi.edu/>.
- [19] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*, volume 1. John Wiley and Sons, 1971.
- [20] J. S. Turner. New directions in communication (or which way to the information age). *IEEE Communications*, pages 8–15, Oct 1986.

Marina Thottan

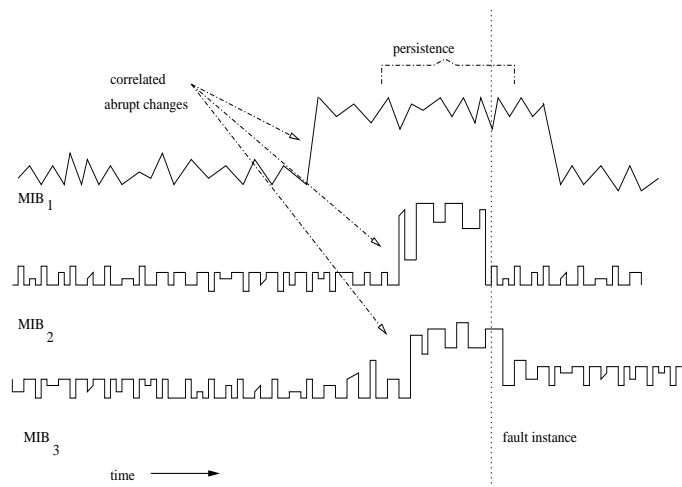
Marina Thottan received her M.S. in Physics from Madras University, India. She received her M.S. in Biomedical Engineering and Ph.D in Electrical Engineering from Rensselaer Polytechnic Institute, USA. Currently she is a Member of Technical Staff in the Department of Network and Service Management at Lucent Technologies, Bell Laboratories.

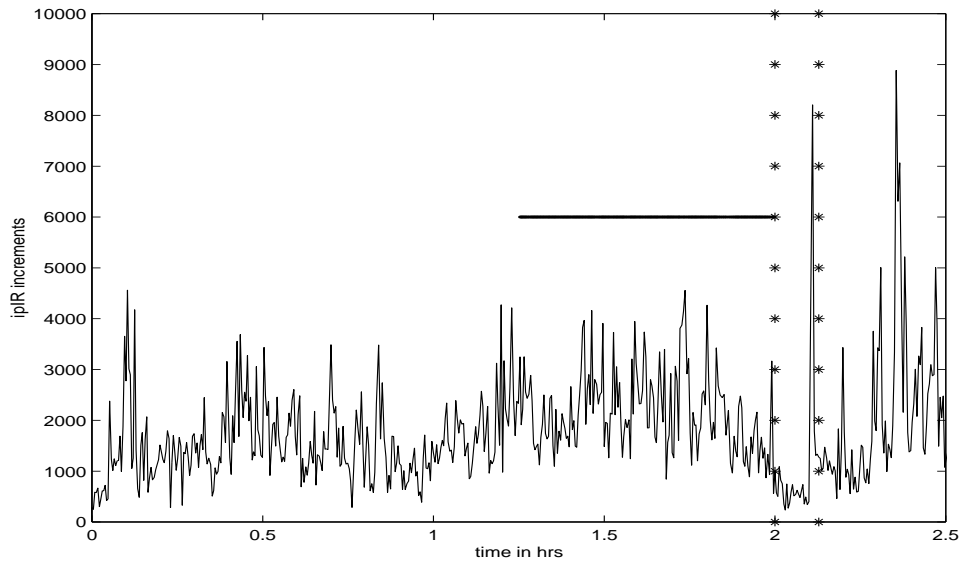
(<http://www.bell-labs.com/user/marinat>)

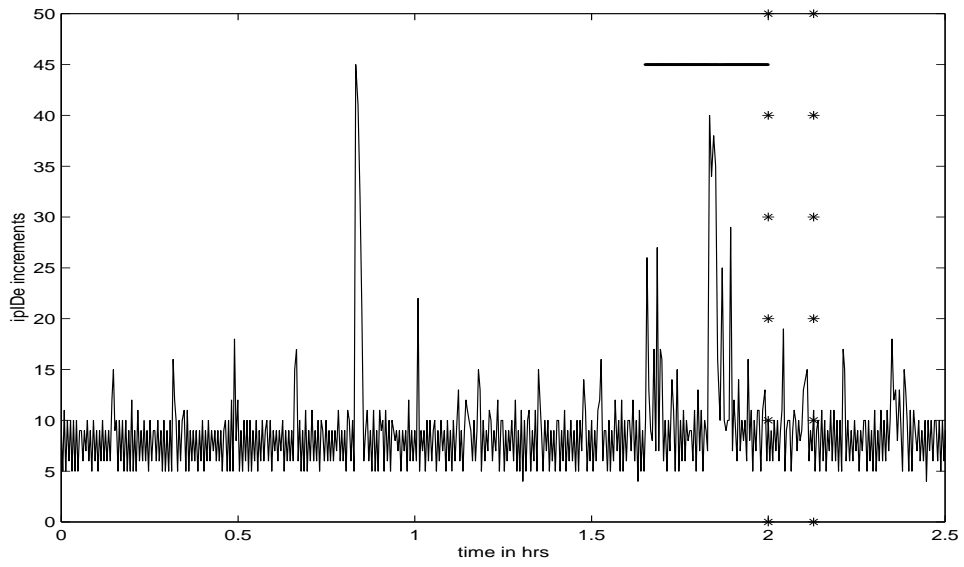
Chuanyi Ji

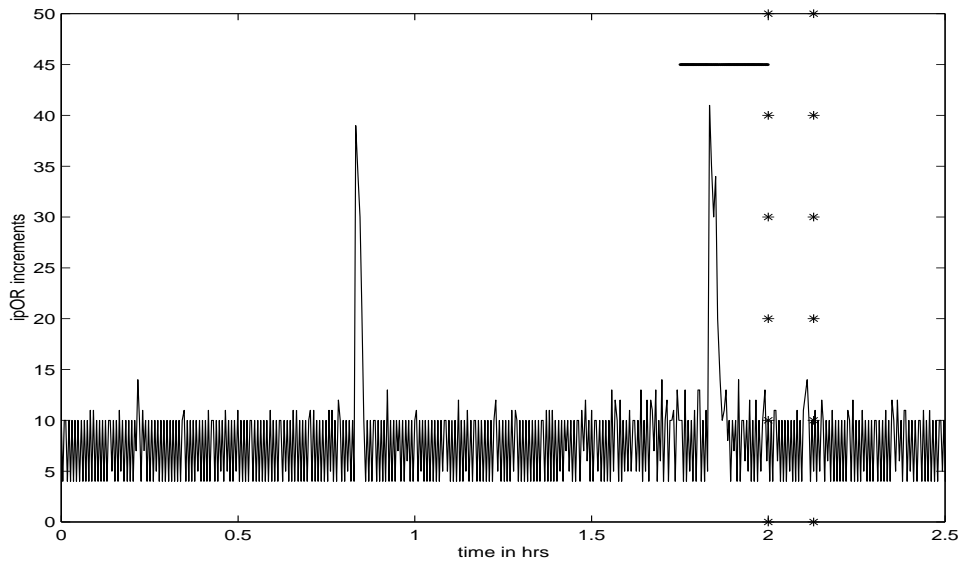
Chuanyi Ji received her BS (with honors) from Tshinghua University, Beijing, Chi na in 1983, an MS from University of Pennsylvania, Philiadelphia in 1986, and a Ph.D from California Institute of Technology, Pasadena, in 1992, all in Electric al Engineering. In November 1991, she joined the faculty at Rensselaer Polytech nic Institute, Troy, where she is now an Associate Professor of Elec-trical, Comp uter and Systems Engineering. Her research interests are in the areas of stochas tic modeling, analysis and control of multi-media traffic, man-agement of hybrid computer communication networks, and adaptive learning systems. Dr. Ji is a reci pient of NSF CAREER award in 1995.

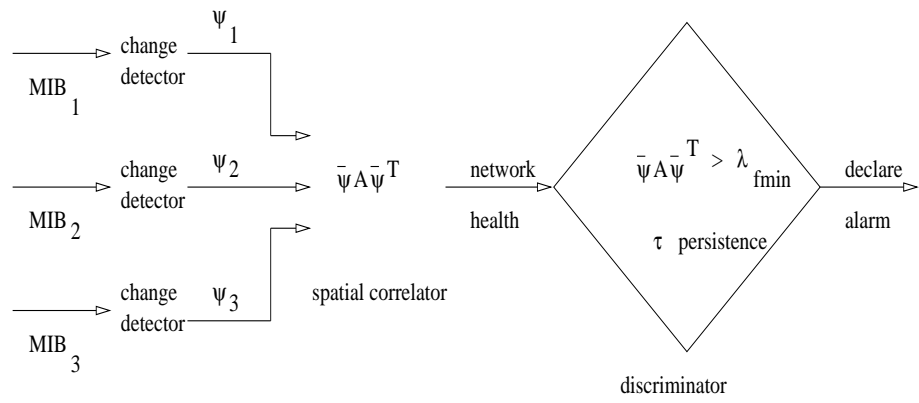
(<http://www.ecse.rpi.edu/homepages/chuanyi>)

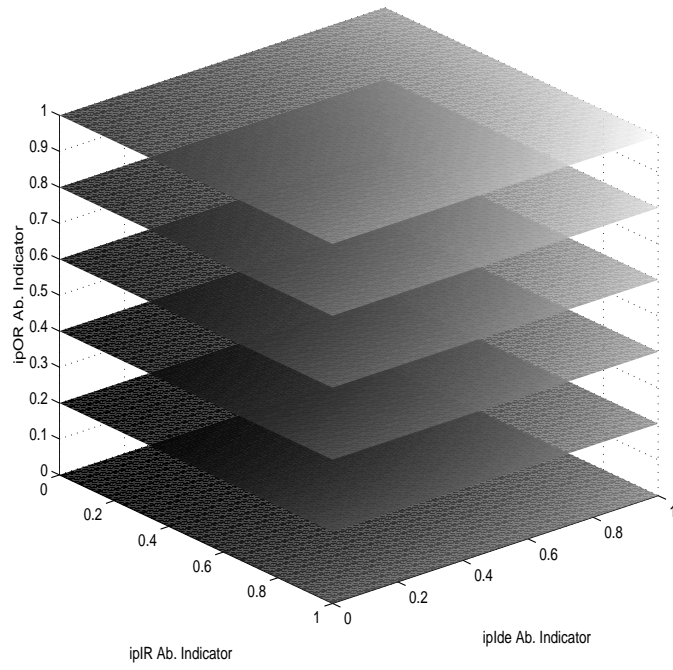


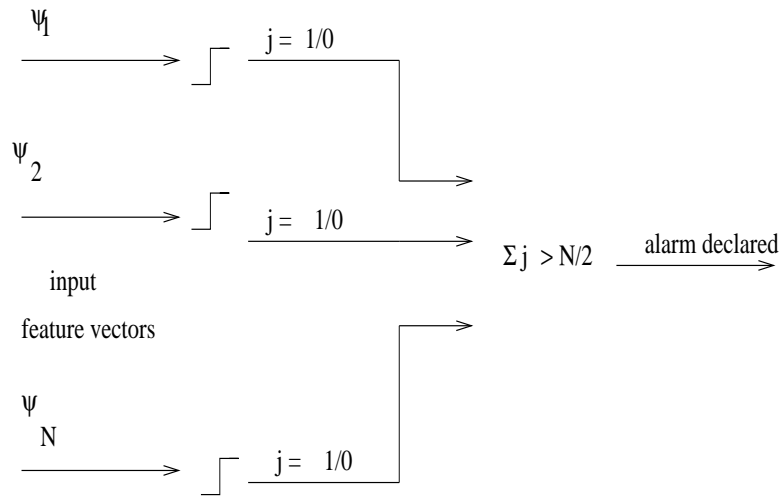


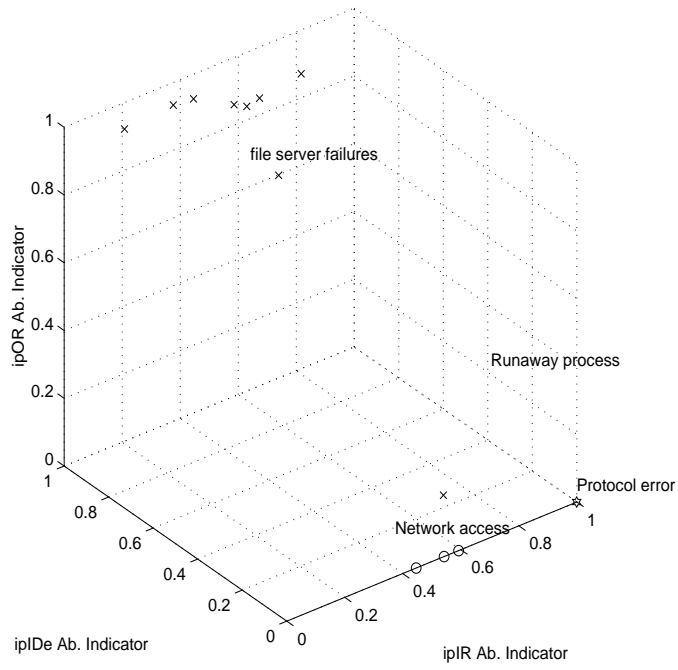












Captions for Figures:

Schematic for Fault model

Trace of *ipIR* Before Fault. Fault Period Denoted by Asterisks. Bold Line Denotes Changes in Time Series Preceding the Fault

Trace of *ipIDe* Before Fault. Fault Period Denoted by Asterisks. Bold Line Denotes Changes in Time Series Preceding the Fault

Trace of *ipOR* Before Fault. Fault Period Denoted by Asterisks. Bold Line Denotes Changes in Time Series Preceding the Fault

Schematic representation of the proposed algorithm

Fault space (shown in lighter shade) embedded in the problem space. The axes indicate the feature vectors (abnormality indicators) obtained from the corresponding input MIB variables

Majority-voting scheme for N input vectors. Σ is the sum of all thresholded feature vectors

Classification of faults using the average (over a 1 hour) of the feature vectors. x: file server failures, o: network access problems, *star*: protocol error, *square*: runaway process

Fault Type	Data Set No	Case No	Prediction Time T_p (mins)	Detection Time T_d (mins)	Mean Time Between False Alarms T_f (mins)
File server	I	1	95	-	700
	II	2	21	-	1032
	III	3	16	-	257
4		-	-		
5		26	-		
IV	6	22	-	1019	
	V	7	15	-	192
VI	8	5	-	184	
	9	5	-		
Protocol implementation error	VII	10	15	-	no other alarms
Runaway process	VIII	11	1	-	235
Network access	IX	12	50	-	286
		13	-	34	
		14	-	12	
Avg.			24.6	23	488
Std.dev			26.9	15.6	370.5

Fault Type	Data Set No	Case No	Prediction Time T_p (mins)	Detection Time T_d (mins)	Mean Time Between False Alarms T_f (mins)
File server	I	1	-	-	105
	II	2	-	29	39
	III	3	-	1	95
		4	-	4	
		5	-	3	
	IV	6	-	1	270
	V	7	-	2	727
	VI	8	-	9	352
		9	-	1	
Avg			-	6.25	265
Std.dev				9.6	233

Fault Type	Data Set No	Case No	Abnormality of <i>ipIR</i>	Abnormality of <i>ipIDe</i>	Abnormality of <i>ipOR</i>
File server	I	1	.5402	0	.1818
	II	2	.5079	.6982	.8180
	III	3	.5669	.9183	.8998
		4	.1497	.9240	.9769
		5	.3698	.9007	.9995
	IV	6	.3061	.9088	.9995
	V	7	.4399	.8099	.9991
	VI	8	.6827	.8248	.9983
		9	.6077	.9142	.9117
Avg.			.4634	.7665	.8650
Std.dev			.1658	.2969	.2638
Protocol implementation error	VII	10	.9999	0	0
Runaway process	VIII	11	.7890	.0909	.5089
Network access	IX	12	.5925	0	0
		13	.4461	0	0
		14	.5424	0	0
Avg.			.5270	0	0
Std.dev			.0744	0	0

Table Captions:
Prediction of failures at the router using the discriminant function scheme
Detection of file server failures at the router using the majority-voting scheme
Abnormality indicators of the feature vectors averaged over an hour prior to the fault