MIB Variable Based Fault Classification: The Next Step Towards Proactive Management
Experience Paper

Marina Thottan
Bell Laboratories
Room 4F-609
101 Crawfords Corner Road
Holmdel, NJ 07733-3030
email:marinat@lucent.com
phone: 732-949-6325

Please use a color printer

Abstract

Faults can be classified using information contained in the MIB variables. It is observed that distinct changes in the MIB data characteristics precede different fault types. Previous work has shown that network faults can be predicted [18] using MIB data. Now with the possibility of classifying faults, network alarms can be associated with specific fault types. Associating a network alarm with a specific network fault is an essential pre-requisite for automated recovery. In our current work we describe the classification of four different faults: network access problems, protocol implementation error, runaway process and file server failures. We also show that a simple discriminant function scheme that accounts for spatial correlations in the MIB data performs better than common majority-voting schemes.

Key-words: fault and performance management, proactive management, fault classification, majority-vote, discriminant functions, statistical methods

1 Introduction

The increased commercial use of the Internet is critically dependent on a reliable network. Hence identification and correction of network faults is an imperative. There are several techniques available today to perform fault detection [6][1]. A major goal of our research is to be able to automate network management through fault prediction, classification and recovery. It has been shown that network faults are preceeded by indications in the MIB variables that allow for prediction[1] [20]. In this work we focus on the classification of faults in an IP network, using the same MIB variable data.

Faults can be classified based on the changes observed in the MIB variables immediately preceeding the fault. This finding has great potential as a means of initiating corrective measures for proactive network management [11][13]. In earlier work we have shown that predictive alarms can be generated using discriminant functions [18]. These discriminant functions are functions of the features derived from the input MIB variables. This is a flexible method since the feature vectors can be tailored to the type of problem under study [4]. The alarms generated correlated well with the faults observed in the data. However, these alarms alone are not sufficient to indicate the type of fault. Once a fault is declared, the alarms have to be associated with a specific fault type in order to implement recovery actions. Fault identification may be done using standard alarm correlation techniques such as finite state machine or constraint satisfaction algorithms [14], [3], [2], [15]. In this work we show, using real network data, that network faults can be identified based on the characteristics of the MIB data immediately prior to the fault. There is no new processing of information required and therefore allows simultaneous declaration and identification of the fault. This provides a significant advantage in terms of the timeliness of recovery measures implemented.

A large number of existing fault detection algorithms use a majority-vote scheme to declare the occurrence of a fault. The majority-vote scheme obscures the details necessary to further classify a network fault due to hard thresholds. In this work we compare our discriminant function scheme with a majority-voting scheme. Using real network data, we found our method performs better in terms of prediction and false alarm rate (the discriminant function predicted most of the faults while the majority-vote only detected one type of fault. The mean time between false alarms using discriminant functions was two times more than with majority-vote.)

The organization of the paper is as follows: In section 2 we discuss some possible choices for feature vectors and provide specific details relating to the choice of feature vectors for this work. Section 2.1 describes the system studied along with the description of the production networks used to obtain data. The metrics used to evaluate management schemes are also described. A discussion on the discriminant function along with the summary of results is presented in section 2.2. The fault classification problem is addressed in section 3. A

[1] with respect to fault labels obtained using *syslog* messages

comparative study of the discriminant function with the majority-voting scheme is provided in section 4. Discussions and conclusion follow in section 5.

2 System Description

The existing management tools provide statistics on a large number of variables that may or may not be relevant to fault detection. Therefore one of the main challenges faced by the research community in this regard is the choice of a single variable or a set of variables that are relevant towards fault detection. Data mining techniques are being used to study different management data bases in order to extract the relevant information [5]. 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. Maxion et al used features such as packet loss, and number of collisions [8]. Trouble tickets can be used as feature vectors in algorithms using artificial intelligence techniques [7],[12], [10]. In our work we use the statistical changes in the Management Information Base (MIB II) variables [9] (which are a part of SNMP).

Since most of the faults of interest are user related we chose those MIB variables that reflect the traffic behavior at a given network node. 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 variables provide a cross-sectional view of the traffic at the network layer. For more details on the choice of these variables refer [17]. The statistical behavior of the change points in the MIB variables are then studied and abnormality indicators[2] are obtained for each of these variables [18]. The value of the abnormality indicator ranges from 0 to 1. A value of 0 corresponds to no change in MIB data behavior and a value of 1 denotes maximal change. Change is measured by comparing adjacent windows of data (approx. 1 hr long) [19]. The abnormality indicators constitute the components of the input feature vector.

2.1 Network Description

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 error, network access problems and runaway processes [17]. Most of these failures were due to abnormal user activity except for the protocol implementation errors. File server failures could result from user behavior such as excessive number of *ftp* requests and a runaway process is an example of high network

[2]The abnormality indicators are obtained using a change detection algorithm. For more details please refer [21]

utilization by some culprit user. However all of these cases did affect the normal characteristics of the MIB data, and impaired the functionality of the network.

The analysis of the two schemes under study consisted of comparing the alarms obtained[3] 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,

$$T_p = T_E - T_a \tag{1}$$

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 scheme under study. The detection time T_d is,

$$T_d = T_a - T_E \tag{2}$$

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.

2.2 Discriminant Functions and Fault Space

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. The discriminant function captures the spatial correlation among the components of the input feature vector $\vec{\psi}$ through the matrix operator A. The feature vector is the abnormality indicators derived from the different MIB variables. The operator is essentially a variant of the correlation matrix ρ [17]. The scheme used is shown in Figure 1. λ is one of the eigenvalues of the matrix

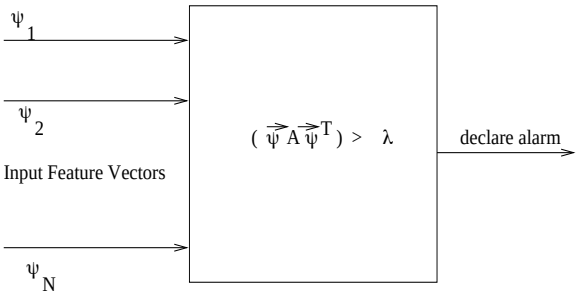

Figure 1: Discriminant function scheme for N inputs using the operator A. λ is the eigenvalue of A.

A. 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

[3] persistence filters incorporated in the interest of robustness [17]

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	1	21	-	1032
	III	1	16	-	257
		2	-	-	
		3	26	-	
	IV	1	22	-	1019
	V	1	15	-	192
	VI	1	5	-	184
		2	5	-	
Protocol imple-Mentation error	VII	1	15	-	no other alarms
Runaway process	VIII	1	1	-	235
Network access	IX	1	50	-	286
		2	-	34	
		3	-	12	
Avg.			24.6	23	488
Std.dev			26.9	15.6	370.5

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

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 [17]. The results obtained using this discriminant function are shown in Table (1). Data set number refers to the different time periods of data collection. Data was collected over a period of ten months. The case numbers in each data set identify the fault instances encountered in the different time periods. *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.

The discriminant function can be used to divide the problem space into a 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 which 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 2. The color scale indicates the gradient in the combined abnormality of the input vectors. Thus the brighter (red) region

which contains the higher values of the abnormality indicators corresponds to the highest abnormal event or a network fault.

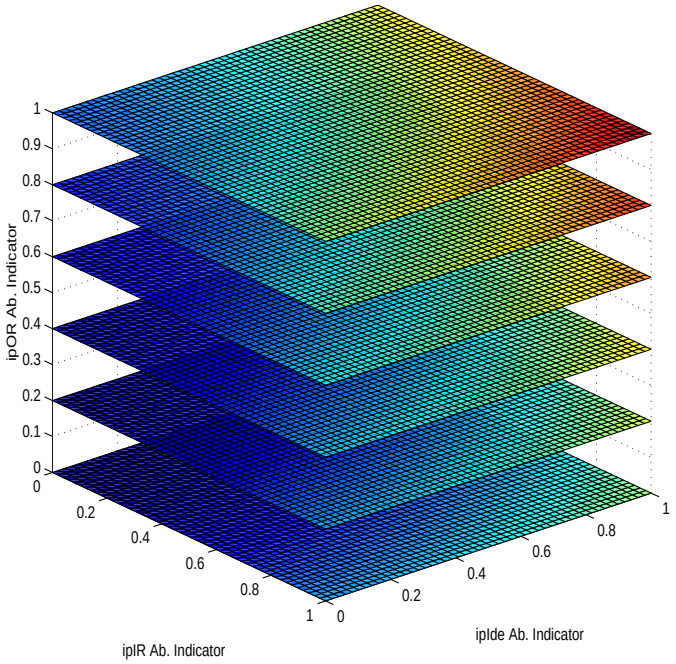

Figure 2: Fault space (shown in red) embedded in the problem space. The axes indicate the feature vectors (abnormality indicators) obtained from the corresponding input MIB variables

3 Fault Classification

Once an alarm is obtained using the discriminant function, we sought to identify the type of impending fault. Using the MIB data 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 (2). This average value is used to locate the fault in the problem space shown in Figure 3.

As shown in Figure 3, 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

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

Table 2: Abnormality indicators of the feature vectors averaged over an hour prior to the fault.

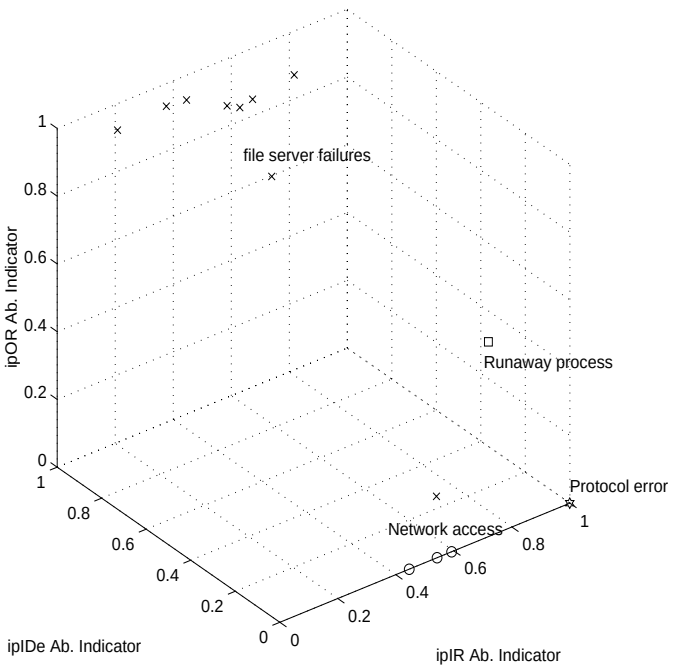

Figure 3: Classification of faults using the average (over a 1 hour) of the feature vectors. x: file server failures, o: network access problems, *star*: protocol error, *square*: runaway process

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.*

4 Comparison of Discriminant Function Method and 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 4. The scheme was implemented on data obtained from the two networks and the results have been tabulated in Table (3).

With our production network data, it was observed that the majority-voting

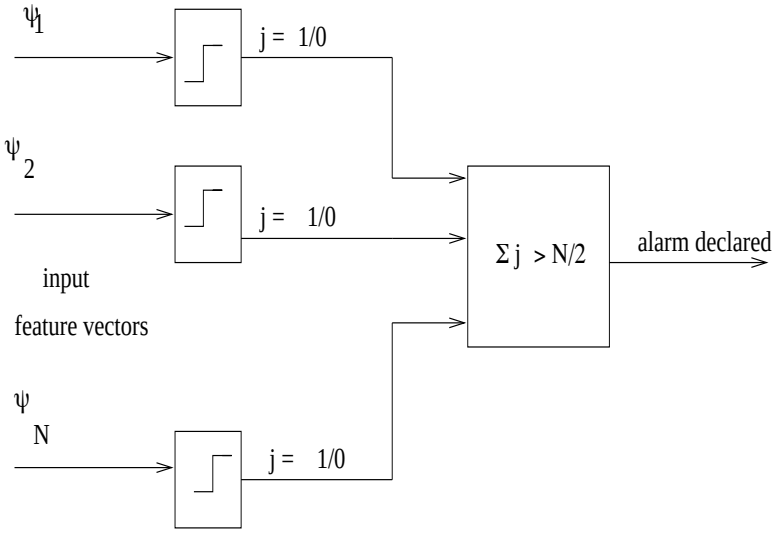

Figure 4: Majority-voting scheme for N input vectors. Σ is the sum of all thresholded feature vectors

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. Furthermore, the discriminant function scheme accounts for the lesser and more subtle spatial correlations among the input feature vectors making it capable of detecting a variety of failures.

The discriminant function was able to predict all of the file server failures. On the other hand, the majority-vote scheme only detected failures at the same 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 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 [16].

In addition to the benefits of prediction, the discriminant function outperforms 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). The discriminant function provides a continuous indicator of network abnormality while the majority-vote scheme

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	1	-	29	39
	III	1	-	1	95
		2	-	4	
		3	-	3	
	IV	1	-	1	270
	V	1	-	2	727
	VI	1	-	9	352
		2	-	1	
Avg			-	6.25	265
Std.dev				9.6	233

Table 3: Detection of file server failures at the router using the majority-voting scheme

gives an on/off output. A continuous indicator is essential to provide trends in availability and reliability information. Thus on comparing Tables (3) and (1) 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 Discussion and Conclusion

In this work we have demonstrated using real network data that the MIB variables show distinctive features prior to a network fault. These distinctive changes 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. As soon as more data becomes available, we hope to confirm our initial findings on the other two fault types as well. 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 proactively managing the network to prevent failures.

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. More research is under way to test the findings in controlled environments and on new network data. Finally, we have shown that using discriminant functions that incorporate the spatial correlations among the MIB variables is significantly better than the majority-vote scheme.

6 Acknowledgments

The author would like to thank Prof. Chuanyi Ji and DARPA for facilitating the early work in this research and Lucent Technologies for providing the enterprise network data. The efforts of D. Hollinger, R. Collins, N. Schimke and C. Hood with the set up of the data collection on the campus network is very much appreciated.

References

[1] A. Bouloutas, G. Hart, and M. Schwartz. On the design of observers for failure detection of discrete event systems. *Network Management and Control, New York: Plenum*, 1990.

[2] G. Jakobson and M. D. Weissman. Alarm correlation. *IEEE Network*, pages 52–59, Nov 1993.

[3] I. Katzela and M. Schwarz. Schemes for fault identification in communication networks. *IEEE/ACM Trans. on Networking*, 3:753–764, 1995.

[4] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[5] 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.

[6] A. Lazar, W. Wang, and R. Deng. Models and algorithms for network fault detection and identification: A review. In *Proc. of IEEE ICC*, 1992.

[7] L. Lewis and G. Dreo. Extending trouble ticket systems to fault diagnosis. *IEEE Network*, pages 44–51, Nov 1993.

[8] R. Maxion and F. E. Feather. A case study of ethernet anomalies in a distributed computing environment. *IEEE Trans. on Reliability*, 39(4):433–443, Oct 1990.

[9] K. McCloghrie and M. Rose. Management information base for network management of tcp/ip-based internets: Mib 2. *RFC1213*, 1991.

[10] C. Melchiors 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.

[11] S. Papavassiliou, M. Pace, and A. Zawadzki. Proactive maintenance tools for transaction oriented wide area networks. *Proceedings of IEEE/IFIP Network Operations and Management Symposium, Honolulu, Hawaii*, pages 847–861, 2000.

[12] 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.

[13] M. A. Rocha and C. B. Westphall. Proactive management of computer networks using artificial intelligence agents and techniques. *Proceedings of IEEE/IFIP Integrated Network Management V, San Diego, USA*, pages 610–623, 1997.

[14] I. Rouvellou and G.W. Hart. Automatic alarm correlation for fault identification. In *Proc. of IEEE INFOCOM*, pages 553–561, Apr Boston USA 1995.

[15] M. Sabin, R. D. Russell, and E. C. Freuder. Generating diagnostic tools for network fault management. *Proceedings of IEEE/IFIP Integrated Network Management V, San Diego, USA*, pages 700–713, 1997.

[16] 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.

[17] M. Thottan. *Fault Detection and Prediction for Management of Computer Networks*. Doctoral Thesis, Rensselaer Polytechnic Institute, Troy, NY, USA., May 2000.

[18] 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.

[19] M. Thottan and C. Ji. Statistical detection of enterprise network problems. *Journal of Network and Systems Management*, 7(1):27–45, 1999. Also available from http://neuron.ecse.rpi.edu/.

[20] M. Thottan and C. Ji. Properties of network faults. *Proceedings of the IEEE/IFIP Network Operations and Management Symposium, Honolulu, Hawaii*, page 941, May 2000.

[21] M. Thottan and C. Ji. Proactive anomaly detection using distributed intelligent agents. *IEEE Network*, pages 21–27, Sept/Oct 1998.

Biography Marina Thottan received her M.S. in Physics from Madras University, India. She obtained an M.S. in Biomedical Engineering and a Ph.D in Electrical Engineering from Rensselaer Polytechnic Institute, Troy, NY. Currently she is working as a member of technical staff in the Department of Network and Service Management at Bell Labs Research. Her research interests are in the areas of adaptive algorithms and management of computer communication networks.