50
Divide and Conquer Technique for Network Fault Management

Kohei OHTA Takumi MORI Nei KATO Hideaki SONE
Glenn MANSFIELD Yoshiaki NEMOTO
Graduate School of Information Science, Tohoku university
Aramaki Aza Aoba, Aoba-ku, Sendai, JAPAN
e-mail: kohei@nemoto.ecei.tohoku.ac.jp

Abstract

From the perspective of fault management, traffic characteristics contain symptoms of faults in the network. Symptoms of faults aggregate and are manifested in the aggregate traffic characteristics generally observed by a traffic monitor. It is very difficult for a manager or an NMS to isolate the symptoms manifested in the aggregate traffic characteristics. Symptoms get obscured by other symptoms. At times there are too many symptoms clouding the symptom space, making the task of symptom isolation practically impossible. In this work we present a powerful technique, the divide and conquer technique, wherein symptoms are iteratively isolated from the aggregate observable. This provides a tractable mechanism for symptom isolation, fault detection and analysis. The symptom isolation technique makes it possible to use a simple thresholding mechanism for detecting abnormalities. We have implemented the system using the popular SNMP-based RMON technology. Using dynamically constructed filters to suppress already detected symptoms in the observed aggregate, fresh symptoms are isolated. Experimental results show a significant improvement in the fault management capability and accuracy.

Keywords

fault management, fault detection, RMON, traffic monitoring

1 INTRODUCTION

In the fault management framework alarms are received by an NMS which has the responsibility of correlating the alarms and locating the faults. A fault may cause several alarms in the network, and several faults may coexist at the same time leading to a cascade of alarms. Mapping alarms to faults is a challenging problem. Several approaches have been suggested e.g. use of coding techniques[KLI95], network configuration information[HOU95][GLN96] etc.. Yet the basic requirements of network fault management[DUP89][STA93] are far from being realized.

Alarms are generated by entities in the network when they sense an abnormality. For example, an agent may be configured to generate an alarm when it sees too many

A. A. Lazar et al. (eds.), *Integrated Network Management V*

ICMP[RFC792]-destination unreachable packets. An alarm may be generated when a router *drops* the default route, or when the operational status of an interface goes *down.*

Detecting an abnormality is a challenging problem. It involves knowing what is *normal* [LAB91]. By comparing an observed value with the normal value an entity may decide whether there is an indication of an abnormality. What is normal for one network may not be so for another network. Dauber[DAU91] suggests a procedure called *baselining* to know what is normal for a network.

In general an NMS collects information about the network. From the perspective of fault management, this information will be analyzed for *symptoms* or indications of abnormal health. The NMS may employ some thresholding mechanism to decide whether the status of a Managed Object (MO) or, the combined status of a set of MOs is indicative of abnormal health or otherwise[GLN92][KOH95]. In the above example, if the number of ICMP destination unreachable packets exceeds a threshold, say *alpha*, an *alarm* or an *event* may be triggered indicating an abnormality in the network and calling for action/intervention. More often than not, this event is the consolidated effect of several symptoms. Each ICMP destination unreachable packet indicates that some destination is unreachable and, as such, is a symptom. It is the task of the NMS to detect the presence of symptoms, isolate and identify each symptom and, diagnose and control the fault indicated by the symptoms.

Symptoms have their own respective characteristics. The *amplitude* of a symptom is a measure of the observable which will be thresholded to decide whether the observable represents an abnormality. The *persistence* of a symptom is a measure of the duration of the symptom and the *frequency* of a symptom is the number of times the symptom has been observed in a given period.

The existence of multiple faults in a network complicates the analysis of symptoms. Some faults and the corresponding symptoms are persistent. If there are persistent symptoms manifested in an aggregated MO, (e.g. ICMP destination unreachable packets), fresh symptoms are likely to get obscured. It is likely that new events are not triggered as the event is already *ON*. Dynamically adjusting the threshold to the "normal" state of the network is a non-trivial problem. Thus some or maybe most of the symptoms occurring in the presence of a persistent symptom are likely to remain hidden, unnoticed. In such situations, the simple thresholding mechanism is a failure at detecting fresh abnormalities and setting new alarms.

Most networks have their own specific eccentricities - which manifest themselves as symptoms. These are *known problems* that are probably under examination. From a managers point of view it would be beneficial to be able to detect fresh problems, i.e., to view the health of a network minus the known problems.

In this paper, we present a powerful technique, the divide and conquer technique, wherein symptoms are iteratively isolated from the aggregated observable. This provides a tractable mechanism for fault detection and analysis. In section 2 we discuss the traffic monitoring approach to fault management, in section 3 we present the divide and conquer technique for fault isolation and management, in section 4 we discuss an SNMP-based practical implementation using widely available RMON-Agent technology, in section 5 we discuss the performance of the proposed technique based on results of experiments on an operational network, followed by conclusion in section 6.

2 TRAFFIC MONITORING FOR NETWORK FAULT MANAGEMENT

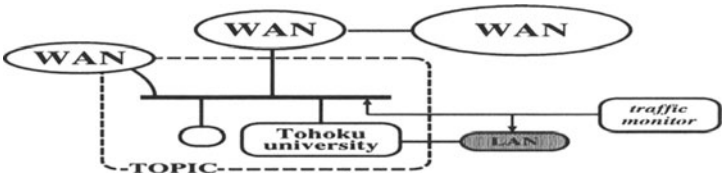

Figure 1 The transit network TOPIC and a stub LAN

2.1 Traffic monitoring

Network management primarily involves monitoring aggregate characteristics. Number of packets, number of collisions, number of broken packets, number of ICMP packets etc. are examples of aggregate characteristics. Aggregate characteristics provide only a macro-view of the network. For example by looking at the number of ICMP destination-unreachable-packets(ICMP-DUR) the NMS can infer that one or more destinations cannot be reached. However for actual fault detection and subsequent diagnosis it is important to know which destination(s) was unreachable and from which source. This micro-view can be obtained by examining the relevant packets in the network.

Special purpose agents, *network monitors*[TCPDUMP][NNSTAT][RFC1757], generally on network entities dedicated to management, provide information about the network traffic as seen by the network entity or monitor. So, a monitor on an Ethernet segment would provide information on all the packets transiting that Ethernet segment. Apart from the macro-view these special agents may also be, in some cases dynamically, configured to provide the micro-view from the network entity i.e. to examine the packets in the network.

This mode of fault management by traffic monitoring has some strong points. It consumes less bandwidth, is more effective and may detect faults from network traffic characteristics, causes of which may lie inside or even outside the management domain.

2.2 Traffic characteristics in a typical network

Traffic characteristics vary from network to network and more so between different types of networks. Characteristics of a transit network are much different from that of a LAN. Fig.1 shows a typical medium-scale network comprising of several WANs and LANs. The area enclosed within the dotted line is the Ethernet backbone of TOPIC(Tohoku Open Internet Community, AS2503)[TOPIC], a network which connects several universities, colleges, museums, and other academic organizations.

Tab.1 shows some statistics of the traffic on the TOPIC network and on another stub LAN connected to TOPIC. The data is obtained by analyzing the packets collected by a traffic monitor over a 24-hour period. The figures of packets dropped are only rough estimates.

Table 1 Traffic characteristics

target network	TOPIC backbone	LAN
number of sender hosts	19028	570
number of receiver hosts	19317	538
number of host pairs	109660	1479
number of ICMP dest unreachable packets	17176	82
number of ICMP dest unreachable sender hosts	482	18
number of ICMP dest unreachable receiver hosts	565	12
number of ICMP dest unreachable host pairs	1194	19
number of lost packets	167975	9501
total packets	38064708	6638313

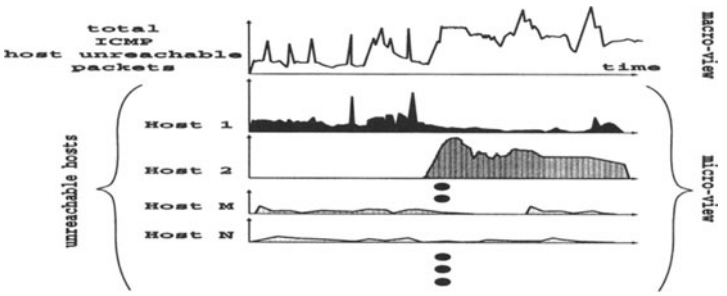

Figure 2 Macro-view & Micro-view

The data shows that for the small scale LAN the traffic is small, the number of sources and destinations are small, so are the number of errors packets e.g. ICMP Destination Un-Reachable packets (ICMP-DUR). In this case, it may be possible to have a fault detection system do an exhaustive examination of each packet in the network.

But the data for the medium-size transit network TOPIC shows very different characteristics. The sample data shows that there were 17176 ICMP-DUR packets during the period of observation. This figure is abnormal. Yet it is only a macro-view and requires the relevant packets to be analyzed. Analysis of the relevant packets showed that 482 destinations were unreachable from 565 sources. In other words, there were 482 symptoms of probable faults in the network during the period of observation.

To examine the time pattern of the symptoms consolidated in the macro-view of ICMP-

DUR packet count we filtered out all the ICMP-DUR packets from the data sample. Then we sorted and separated the different *Destination-IP addresses* in the filtered sample. Each unreachable destination represents a symptom. The various symptoms are shown in fig.2. It is evident from the analysis that the aggregated macro-view represented by the total number of ICMP-DUR packets is a consolidation of several symptoms - each representing the unreachability of a different host, shown in the graphs of fig.2.

The above analysis was done offline. The point to be noted is that, it is impractical to examine the internals of each packet for online-management purposes even with dedicated machines[KIM93][STA93].

In Fault Management it is necessary to use the macro-view to select a potential problem spot, an event. And, then focus on the micro-view of that event.

3 FAULT DETECTION SYSTEM WITH DIVIDE AND CONQUER TECHNIQUE

3.1 Fault detection model

In our model (fig.3) of fault detection there are essentially three parts. An event, E, is detected in the Event-detection phase. This event is essentially a macro-view that indicates the presence of one or more symptoms of faults. The NMS focuses on the set of symptoms S', $S' \subseteq S$, that are likely to have triggered the event, and finds S'' the set of symptoms that did trigger E. The NMS then proceeds with the diagnosis of the symptoms and corrective procedures for the corresponding faults. Further, having isolated the symptoms, it filters out these symptoms from subsequent observations. Detailed explanation of each phase follows.

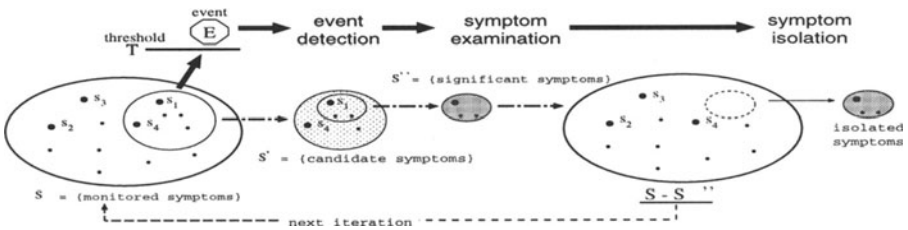

Figure 3 Fault detection model

Event detection

Ideally an event E is triggered when, the amplitude, $a(s_i)$, of a symptom s_i exceeds some threshold t_i for s_i.

$$a(s_i) > t_i \implies E = true$$

However, for practical purposes, instead of applying the threshold to individual symptoms s_i, it is more effective to to apply the threshold to an aggregated set of symptoms $S = \{s_i\}$.

$$\sum_i a(s_i) > T, s_i \in S \implies E = true$$

It is clear that two or more symptoms may cooperate to trigger an event. Thus the NMS has to carry out the process of identifying the symptoms that did trigger the event. Further, the threshold T is in general set for a single symptom. This threshold when applied to the aggregated amplitude of several symptoms is small and, is almost always exceeded.

Symptom examination

Though several symptoms S', $S' \subseteq S$, cooperate to trigger an event, all symptoms are not significant. The NMS will use some mechanism to isolate the significant symptoms S'', $S'' \subseteq S'$.

$$S'' = \{s_i | s_i \text{ is significant}, s_i \in S'\}$$

The significance of a symptom may be gauged by one or more characteristics of the symptom e.g amplitude, frequency, persistence or by specific knowledge about the symptom. The symptoms are then used for diagnosis.

Symptom isolation

To avoid the preponderance of already detected, persistent symptoms , the set of significant symptoms S_t'' from the Symptom examination phase in the t^{th} cycle are suppressed in the subsequent event detection phase. Thus the symptom space S_{t+1} in the $t+1^{th}$ cycle is given by

$$S_{t+1} = S_t - S_t''$$

3.2 The Divide and Conquer technique

Fig.4 shows the basic components for our proposed event-driven, dynamic symptom isolation system. First, an event is detected by monitoring aggregated characteristics. Next, symptoms that triggered the detected event are analyzed. Significant symptom(s) are isolated from the aggregated characteristics of the subsequent cycle, in the last component. This cyclic process isolates the symptoms iteratively.

4 IMPLEMENTATION OF THE DIVIDE AND CONQUER TECHNIQUE

The authors implemented and experimented with the Divide And Conquer Technique. The implementation was on the SNMP platform and used the RMON-MIB[RFC1757] for traffic analysis. The RMON-MIB is a powerful tool for traffic analysis and provides

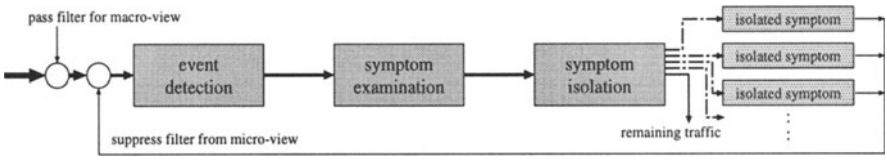

Figure 4 Basic components of iterative isolation

mechanisms for applying filters for analyzing traffic, and to configure alarms and events, by using SNMP-set and get constructs.

An NMS in general needs to watch out for several types of events. The data for some of these events are readily available from counter values of predefined MOs. For the remaining events the RMON Agent needs to be configured to generate the data by analyzing the traffic. This generally involves applying filters and counting packets that pass the filter. Though theoretically an infinite number of filters may be applied and a watchout may be maintained for all possible events, there is a practical limit to the number of filters that can be used and hence on the number of events that can be watched out for simultaneously. To get around this problem we used sampling techniques. It essentially involves time-division multiplexing - in each time slot a particular event is checked for by applying the appropriate the set of *pass* filters.

Though the NMS in general scanned for several events, in the following we will concentrate on events due to the ICMP-Host-Unreachable(ICMP-DUR) packets. Since there is no ICMP-DUR counter readily available, we had to configure the RMON-Agent to generate the count. This was done in the **Event Detection** phase by sampling the traffic and filtering for ICMP-DUR packets. In this phase *suppress* filters are applied to suppress known symptoms (detected in the **Symptom Examination** phase), if any. The sampling was necessitated for overall performance purposes. An event would be triggered when the number ICMP-DUR packets exceeded the threshold.

On being notified of an event (by a trap) the NMS would begin the **Symptom Examination** phase. The NMS starts off a packet-capture process. All ICMP-DUR packets are captured and analyzed by the NMS. The analysis is carried out by sorting the packets by the IP-address of the unreachable destination (in the IP-header that is sent as data in the ICMP packet). Each unreachable destination is a symptom. The symptoms are then arranged in frequency of occurance. In the **Symptom Isolation** phase, the Top N symptoms greater than the threshold are selected as significant symptoms indicating real faults, needing diagnosis. Filters are made corresponding to these symptoms and are used as suppression filters in the **Event Detection** phase.

4.1 Structure of the system

An outline of the implementation is shown in fig.5. In the figure, rounded rectangles represent the RMON-MIB MOs, shaded areas represent the components of the concept, and shaded rectangles represent sub-processes. Arrows on continuous lines represent packet flows, arrows on broken lines represent instructions from the NMS.

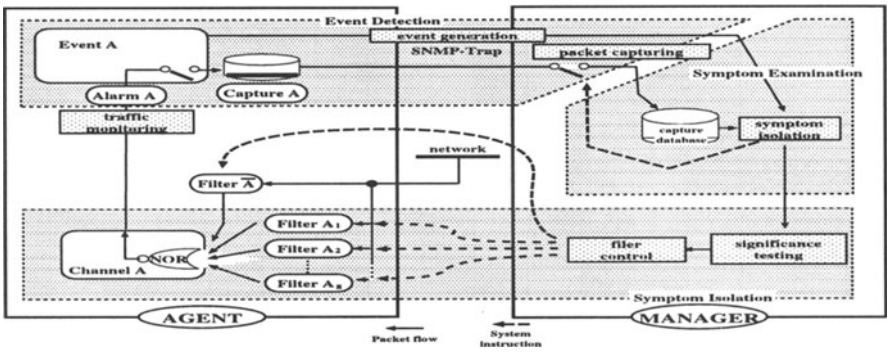

Figure 5 Implementation using RMON technology

4.2 Event detection

In this phase traffic monitoring and event generation takes place. The manager sets the `filter` and `channel` MOs for a particular macro-view or event, Filter A and Channel A in fig.5. The agent monitors the target traffic which passes through the pass-filter for the macro-view covering the set of symptoms S. An alarm A is generated when the monitored statistic is over the predefined threshold T. The alarm A in turn triggers an event E which starts the packet capture process in the `capture` MO and sends a notification about the event to the NMS. (Via a trap).

4.3 Symptom examination

The Symptom examination process comprises of two sub-processes, packet capturing and symptom analysis. Once the NMS receives notification of the event E, it will want to examine the symptoms that are manifested in the traffic characteristics. For this purpose, the NMS needs a trace of the traffic. The NMS obtains the trace from the RMON-agent by polling the corresponding `capture` MO, Capture A in fig.5. From the trace the NMS finds the symptoms S' that cooperated to trigger the event.

4.4 Symptom isolation

In this phase the NMS picks the significant symptoms S'' from S', using some criteria, for diagnosis. The NMS decides whether a symptom is significant or not by analyzing its characteristics. A simple decision can be made based on frequency. The symptom that is occurring frequently, probably needs attention and is significant. On the other hand there may be a database of symptoms which may be looked up to ascertain the severity of an event. For example, if a DNS server has become unreachable - it is certainly a severe fault that may affect the network users. In the pilot implementation, we used the Top-N method to determine whether a symptom is significant. The symptoms are sorted

by frequency, and the top N symptoms are considered to be significant. The figure N is implementation specific and in general depends on the number of available filters, etc.

Next, the manager sets the filters corresponding to the isolated symptoms to suppress these already-detected symptoms in the subsequent traffic. The reduced set of symptoms are

$$S_{t+1} = S_t \cap \overline{S''}$$

In the RMON architecture the above translates to

$$S_{t+1} = \overline{\overline{S_t} \cup S''}$$

In fig.5, Filter $\overline{A}$ corresponds to $\overline{S_t}$ and Filters $A_1 \sim A_N$ correspond to the isolated symptoms in S''. Channel A corresponds to the RMON-MIB channel object where the Filters A, $A_1 \sim A_N$ are combined. When the threshold for a particular macro-view is crossed an alarm is triggered. This alarm is configured in the RMON-MIB alarm MO, Alarm A in fig.5. The alarm A in turn is configured to trigger an event MO, Event A, which sets off a trap to the NMS notifying the NMS of the event and, starts the packet capture in the MO Capture A.

5 PERFORMANCE EVALUATION

5.1 Fault detection by monitoring ICMP packet flow

To evaluate the proposed system, we experimented on fault detection by monitoring the ICMP traffic in the network. ICMP belongs to the TCP/IP protocol suit and is primarily used between network entities to notify problems about network reachability, congestion, packet loss, etc.

5.2 System configuration for evaluation

The experimentation and system evaluation was carried out on the TOPIC backbone network which is an Ethernet(fig.1). A dedicated RMON-Agent was connected to the Ethernet. We concentrated on the ICMP destination unreachable packets.

The traffic was sampled every 5 minutes for a duration of 1 minute. For traffic analysis the sampled traffic of the most recent 180 minutes was used. Furthermore, since the RMON device we used offered a maximum number of 10 filters - we employed 8 for symptom (suppression) filters and 2 for the event (pass) filters.

5.3 Results and Observations

Order from Chaos: tractability of event monitoring

Fig.6-a shows the total number of ICMP-DUR packets. A very large number of ICMP-DUR packets are observed and there is a wild fluctuation in the number. It is difficult to

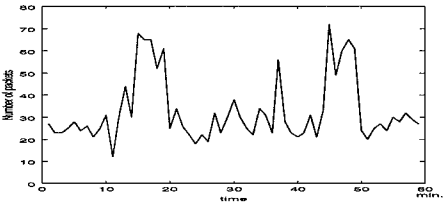

Figure 6-a ICMP-DUR pkts (raw)

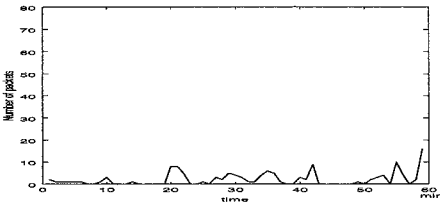

Figure 6-b after symptom isolation

Figure 6 Number of ICMP-DUR pkts/minute

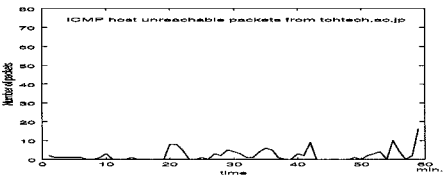

Figure 7-a Symptom #1

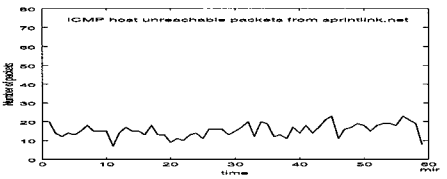

Figure 7-b Symptom #2

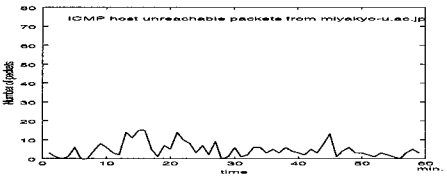

Figure 7-c Symptom #3

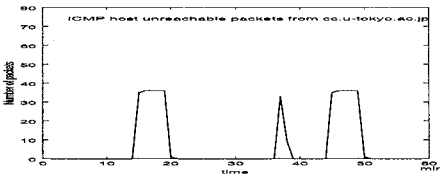

Figure 7-d Symptom #4

Figure 7 Samples of divided traffic

make much sense from the graph apart from forming the general idea that there are "too many" ICMP-DUR packets. Also, setting an event threshold at any sensible value seems to be impossible ! On the other hand fig.6-b shows the count of ICMP-DUR packets after the symptoms have been isolated. This graph shows that the problem-space (number of unresolved or unknown symptoms of faults present in the traffic) is well within control. There are no wild fluctuations. Occasional peaks do get resolved i.e. the corresponding symptoms do get identified and isolated quickly.

In fig.7 we show the characteristics of some of the isolated symptoms corresponding to the one hour period in fig.6. (The order of the graphs is insignificant). From the graphs it is evident that for detection purposes symptoms may be individually subjected to a uniform threshold. This is a very significant result corroborating our claim of the usability of a simple thresholding mechanism for fault detection and management.

Symptom detection capability

To measure the impact on symptom detection capability we compared the performance of our system which uses the the divide and conquer technique with a system that doesn't.

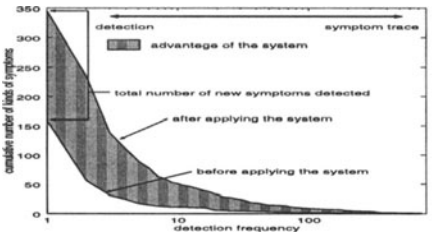

Figure 8 Overall evaluation

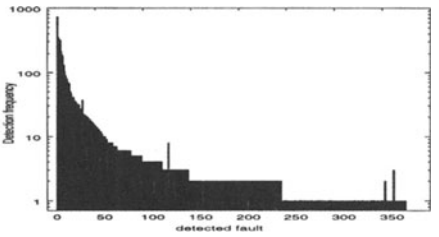

Figure 9 Detection accuracy

Both systems used identical RMON-Agents to set appropriate filters for alarms, to obtain event information and to carry out packet capture. Both systems analyzed the captured traffic to identify symptoms and employed similar thresholding mechanisms.

The results are shown in fig.8. The ordinate represents the number of symptoms, and the abscissa represents the minimum number of times the corresponding symptoms were detected. We can see that the number of symptoms which were detected more than 10 times using the divide and conquer technique, is 50 and, 12 otherwise. Of particular interest is the number of symptoms detected (at least once). It is evident that using the divide and conquer technique the symptom detection capability is significantly ($> 200\%$) improved.

Symptom detection accuracy

Fig.9 compares the accuracy with which the two systems detect symptoms. The figure shows the number of times each symptom is detected by the system using the divide and conquer mechanism and the one that doesn't. The dark and the gray bars represent the number of times the corresponding symptom has been detected by the divide and conquer system and the traditional system, respectively. The symptoms are ordered by frequency of occurance (as seen by the system that employs the divide and conquer technique). It is clear that the divide and conquer mechanism does increase the accuracy of symptom detection significantly. In the case of some symptoms it does appear that divide and conquer mechanism is less accurate. Particularly towards the tail end of the spectrum. It may be noted that there is an inherent latency in the symptom detection mechanism. It takes a finite amount of time after the event detection for the system to start the capture process, and then do the symptom examination. Symptoms which have very short persistence, disappear in this time and thus elude detection. This syndrome is present in both systems. Yet, since the two systems are not synchronized in the strict sense, some symptoms elude one system and are caught in the other. The concentration of symptoms that eluded the divide and conquer technique based system at the tail of the spectrum, is attributable to the ordering of the symptoms in the figure.

6 CONCLUSION

In this paper, we have focussed on the issue of network fault management by traffic monitoring. We have proposed a simple and powerful technique, the divide and conquer

technique which makes the problem of detecting symptoms of faults in the network more tractable. By applying this technique, one can use simple thresholding mechanisms for setting alarms and detecting events in the network. We have implemented the system using standard network management protocols and technology. We have compared its performance with conventional systems. The divide and conquer mechanism does enhance the performance significantly- the spectrum of symptoms detected are broadened and the accuracy with which symptoms are detected is increased.

REFERENCES

[KLI95] S. Kliger et.al "A coding approach to Event Correlation ", Proceedings , Fourth International Symposium on Integrated Network Management, 1995.

[HOU95] K. Houck et.al "Towards a Practical Alarm Correlation System ", Proceedings , Fourth International Symposium on Integrated Network Management, 1995.

[GLN96] G.Mansfield, M.Ouchi, K.Jayanthi, Y.Kimura, K.Ohta, Y.Nemoto, "Techniques for automated Network Map Generation using SNMP", Proc. of INFOCOM'96, pp.473-480, March 1996.

[DUP89] Dupuy.A, et.al " Network Fault Management :A User's View", Proceedings , First International Symposium on Integrated Network Management(May), 1989.

[STA93] William Stallings: "SNMP, SNMPv2, and CMIP - the Practical Guide to Network Management Standards -", Addison-Wesley Publishing Company 1993.

[RFC792] SJ. Postel "Internet Control Message Protocol ",RFC 0792 09/01/1981

[LAB91] Lee LaBarre "Management by Exception: OSI Event Generation, Reporting and Logging ", Proceedings , Second International Symposium on Integrated Network Management, 1991.

[DAU91] Steven M. DAUBER, "FINDING FAULT", BYTE Magazine, March 1991.

[GLN92] G.Mansfield, et.al "An SNMP-based Expert Network Management System",IEICE Trans. COMMUN.,Vol.E75-B,NO.8, pp.701-708, August 1992.

[KOH95] K.Ohta, et.al "Configuring a Network Management System for Efficient Operation", International journal of network management, Vol.6, No.2 March-April 1996.

[TCPDUMP] V. Jacobson, C.Leres, and S.McCanne, "tcpdump", available via a-FTP to ftp.ee.lbl.gov, June, 1989.

[NNSTAT] R.R. Braden and A. Deschon. NNStat, "Internet statistics collection package. Introduction and User Guide", Technical Report RR-88-206, ISI, USC, 1988. Available for a-ftp from `isi.edu`.

[RFC1757] S. Waldbusser, "Remote Network Monitoring Management Information Base", RFC 1757 02/10/1995.

[TOPIC] Description about TOPIC at: `http://www.topic.ad.jp/`

[KIM93] Kimbery C Claffy and George C. Polyzos "Application of Sampling Methodologies to Network Traffic Characterization", IEEE SIGCOM'93 1993.

[RFC1157] M. Schoffstall et al.: "Simple Network Management Protocol SNMP", RFC1157,1990.

7 BIOGRAPHY

Kohei Ohta received his B.E and M.S degrees in information engineering from Tohoku University in 1993 and 1995, respectively. Now he is a doctoral student at Graduate School of Information Sciences of Tohoku University. He has been engaged in research on network management.

Takumi Mori is a senior of information engineering of Tohoku University. He has been engaged in research on network management.

Nei Kato received his M.S. and the Dr.Eng. degrees in information engineering from Tohoku University in 1988 and 1991, respectively. Now he is an associate professor at Graduate School of Information Sciences, Tohoku University. He has been engaged in research on pattern recognition, neural network and computer networking. Dr. Kato is a member of IEEE and the Information Processing Society of Japan.

Hideaki Sone received his B.E. in Electrical Engineering and M.E. and Doctoral Degree in Electrical Communications from Tohoku University, Sendai, Japan. He is an Associate Professor in Research Institute of Electrical Communication, and is working with Computer Center, Tohoku University. His main research interest lies in the fields of communication systems and instrumentation electronics. Dr. Sone is a member of the IEICE, the IEEJ, the SICE, the IPSJ, and the IEEE.

Glenn Mansfield obtained his Master's degree in 1977 from Indian Institute of Technology, Kharagpore, India in the field of Nuclear and Particle Physics followed by his Masters in Physical Engineering in 1979 from Indian Institute of Science, Bangalore, India. He obtained his Ph.D. specializing in Logic programming, from Tohoku University, Japan. He is currently chief scientist at Sendai Foundation for Applied Information Sciences, Japan. His areas of interest include expert systems, logic programming, computer networks and their management, use of the Internet for education. He is a member of the Internet Society, the ACM, the IEEE and the IEEE Communications Society.

Yoshiaki Nemoto received his B.E. and M.E. and Ph.D. degrees from Tohoku University, Sendai, Japan, in 1968, 1970 and 1973 respectively. He is a professor with Graduate School of Information Sciences, Tohoku University. He has been engaged in research work on microwave networks, communication system, computer network system, image processing and handwritten character recognition. Dr. Nemoto is a member of the IEEE and the Information Processing Society of Japan.