An Intelligent Mobile Agents-Based Architecture for Network Fault Detection*

ZHANG Pu-han, SUN Yu-fang

(*Institute of Software, The Chinese Academy of Sciences, Beijing* 100080, *China*)

E-mail: phzhang@sonata.iscas.ac.cn; yufang@admin.iscas.ac.cn

http://www.iscas.ac.cn

Received May 22, 2001; accepted October 23, 2001

Abstract: The increasing complexity of modern network has motivated the development of different fault detection and isolation approaches for the purpose of supervision. Currently, packet monitoring has become a standard technique in network fault detection, but when applied to a large-scale network it yields a high volume of packets. To overcome this problem, some techniques are proposed. However, the proposed techniques are based on popular SNMP agent and RMON technology, which are characterized by centralization, some inherent known problems are the lack of scalability, complexity to configure, network congestion and not strong local processing ability. This paper proposes a new method for fault detection and isolation based on intelligent mobile agents. To measure the impact on fault detection capability, a monitoring and fault detection experiment is successfully implemented, and the authors compare the number of symptoms detected and traffic of the system with a traditional system using RMON agents. Experimental results show that the fault detection capability using the proposed approach is significantly improved.

Key words: network fault detection; intelligent mobile agent; ICMP packet monitor; network management

As networks become larger and more complex, the need for advanced fault management capabilities becomes critical. Faults are unavoidable in large and complex communication networks, but quick detection and identification can significantly improve network reliability. A single fault in a large communication network may result in many fault alarms, making the isolation of the primary source of failure a difficult task. The problem becomes worse in the case of multiple faults.

A fault is an abnormal condition (probably due to hardware and/or software error). Furthermore, abnormal usage which generally indicates that the network's security is breached also needs to be monitored and attended to. And abnormality, whether it is due to a fault or to an attack, needs prompt management attention and so should be detected at the earliest possible instance.

Fault detection forms the base for fault management. It involves knowing what is normal[1]. By comparing an observed value with the normal value an entity may decide whether there is an alarm of an abnormality.

In general, alarms are generated by entities in the network when they sense an abnormality. In alarm-based fault detection systems, often a single fault will cause a large number of alarms. Also, several faults may coexist causing a cascade of alarms. These alarms must be correlated to pinpoint their causes so that problem can be handled

* Supported by the National Natural Science Foundation of China under Grant No.69983009 (国家自然科学基金)

ZHANG Pu-han was born in 1970. He is a Ph.D. candidate at the Institute of Software, CAS. His research interests are computer network and mobile agent. **SUN Yu-fang** was born in 1947. He is a professor and doctoral supervisor of the Institute of Software, CAS. His current research areas are database, Chinese information processing, operating system and computer network.

effectively.

Mapping alarms to faults is a challenging problem. Several approaches have been suggested, e.g. use of coding techniques[2], network configuration information[3], etc. Yet the basic requirements of network fault management are far from being realized. Packet monitoring[4] is by far the richest source of network information on hosts, application, fault type and so on. For example, in a system based on SNMP, an agent may be configured to generate an alarm when it sees too many ICMP[5] packets. If the alarm is triggered, the agent collects ICMP packets, and diagnoses what happened in the network from collected packets. However, the multitude of packets gathered from a network can easily obscure the real cause of the fault. Furthermore, multiple faults may coexist. So, to determine faults from observed packets, special techniques are required. As popular approach, packet monitoring-based fault detection systems have been proposed[6~8].

But the proposed techniques are based on popular SNMP[9], RMON(Remote MONitoring)[10] technology, which are characterized by centralization. A network manager residing on a central station contains most of the management logic and processes the data collected from physically distributed agents. The agents are rigid servers that are closely associated with the hosting network components. The involved management paradigm concentrates most of the processing into a single manager that usually interacts with a large number of agent servers. Hence, they do not provide the strong local processing ability and the required scalability that is needed in today's predominantly complex networks due to the large number of network components, vast topologies, unpredictable network dynamics, etc.

The basic idea to solve these problems of central structure and a high volume of packets is to bring management intelligence and mobility as close as possible to the managed resources. One of the most prominent techniques providing a solution is management by mobile agent[11]. It represents a clear effort towards decentralization and increased flexibility of management functionality. Instead of the traditional methods of exchanging client/server messages, the management station can specify a task to be carried out by locating a program (a mobile agent) on involved devices, where the actual execution of the task takes place. The agents have the ability to travel from one node to another and can benefit from the integration of artificial intelligence technologies, which capture the human manager expertise in solving network problems.

In this paper, we present an intelligent mobile agents-based architecture to deal with the problem of network fault detection by using ICMP packet monitoring technique and implemented it by using java technology and aglet[12] which is a mobile agent system from IBM. In Section 1 we discuss the intelligent mobile agent and the packet monitoring approach to fault detection, in Section 2 we present the intelligent mobile agents-based architecture for fault detection, in Section 3 we discuss an practical implementation using java and aglet, and the performance of the proposed architecture based on the experimental results, followed by conclusion in Section 4.

1 Background

1.1 Intelligent mobile agent

An agent can be referred as a component of a software and/or hardware process that is capable of acting on behalf of a user or in response to predefined conditions.

The primary characteristics of an intelligent mobile agent comprise the degree of operational autonomy, intelligence, the mobility and communication.

For an agent to be autonomous, it must be a separate thread or process that waits, ready to respond to a user request or a change in the environment.

A mobile agent can migrate from one location to another assuming the destination gives its approval. The

primary requirement for mobile programs is the ability to save the state of the running process, transport it and then resume where the process left off.

The level of intelligence in intelligent agents can range from hard-coded procedures to sophisticated reasoning and learning capabilities. The specific functions and requirements of an intelligent agent are the prime determinant of which techniques should be used.

Combining these three properties (autonomy, mobility, and intelligence) is a software entity coined an Intelligent Mobile Agent.

When agents communicate with one another, they can either talk directly provided they speak the same language or talk through an interpreter or facilitator. The Knowledge Query and Manipulation Language (KQML)[13] provide a framework for a set of independent agents to communicate and cooperate on a problem. In this paper, agents communicate with each other by using KQML which is simple and enough to finish the work in our system.

1.2 ICMP packet monitoring for network fault detection

Fault detection primarily involves monitoring traffic characteristics, i.e. number of packets, number of collisions, number of broken packets, number of ICMP packets etc. are examples of characteristics. Packet monitoring is by far the richest source of network information on fault type. In this paper, we focus on ICMP packet monitoring for fault detection. ICMP belongs to the TCP/IP protocol suite and is primarily used between network entities for sending notifications of problems related to network reach ability, congestion, packet loss, etc. The implementation of ICMP is mandatory for TCP/IP systems. Thus monitoring ICMP packets is the most suitable approach for fault detection.

ICMP is used for error messages, and other messages intended for the TCP/IP software itself, rather than any particular user program. For example, if you attempt to connect to a host, your system may get back an ICMP message saying “host unreachable”. ICMP can also be used to find out some information about the network. Table 1 shows the messages notified by ICMP packets. Now, we pay attention to ICMP destination unreachable message. Table 2 shows typical error codes delivered by a destination unreachable message[14]. Codes 0, 1, 4, and 5 may be received from a gateway. Codes 2 and 3 may be received from a host.

Table 1 Error messages notified by ICMP

Type	Function
8/0	Echo Request/Reply
3	Destination Unreachable Message
	Source Quench Message
4	Redirect Message
5	Router Advertisement/
9/10	Solicitation Message
11	Time Exceeded Message
12	Parameter Problem Message
13/14	Timestamp Request/Reply
15/16	Message
	Info Request/Reply Message
17/18	Address Mask Request/Reply

Table 2 Typical ICMP destination unreachable code

Code	Function
0	Net Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation Needed and DF Set
5	Source Route Failed
13	Prohibited by filtering

The Internet Protocol is not designed to be absolutely reliable. The purpose of these ICMP control messages is to provide feedback about problems in the communication environment. Thus ICMP packets monitoring is the most suitable approach for fault detection. For a small scale LAN the traffic is small, the number of source and destinations are small, so is the number of ICMP packets. In this case, it may be possible to have a fault detection system do an exhaustive examination of each packet in the network. But if the traditional ICMP-packet

monitoring-based fault detection technique is applied to a large-scale network like a transit network, it is difficult to diagnose the actual problem for the following reasons. (1) In a large-scale network, a single fault generally results in a number of packets not reaching their destination. Each packet gives rise to an ICMP error packet. For this reason, it is difficult to find the real fault from many packets. (2) If several faults coexist at the same time, complex packet flows are observed and it can be difficult to determine how many faults have occurred.

So, to overcome this problem to determine faults from observed packets, we design the following intelligent mobile agents-based architecture.

2 The System Architecture

The comparison of structure of the system based on intelligent mobile agents with the one of traditional system based on SNMP and RMON technology is shown in Fig.1.

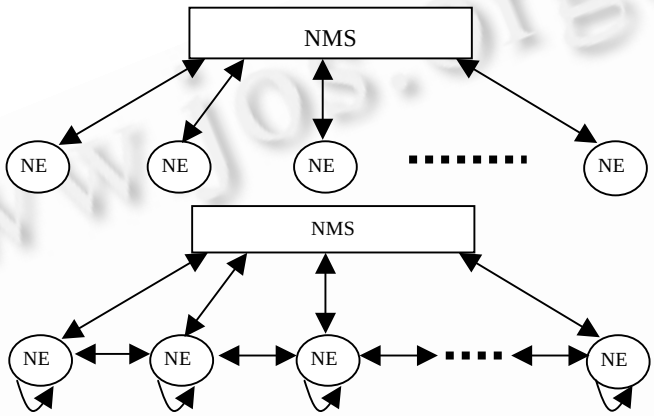

Fig.1 RMON vs IMA

A traditional centralized Network Management Station (NMS) uses the SNMP protocol to communicate with the RMON agent, and uses RMON agent to monitor the packets. Detection and capturing was implemented using the alarm and capture function of RMON. The RMON agent observed ICMP packets, captured the relevant packets and notified and manager in the NMS. When the manager received notification from RMON agent, the manager fetched the captured packets from RMON.

RMON agents act merely as packet collectors that provide information only when they receive a request from the manager. Such agents cannot perform processing on their own due to their design limitations and processing inflexibility. Many of the traditional systems reside at remote sites and polling over wide area links quickly increases operating costs. Even on high-bandwidth LANs that are being increasingly burdened with delay sensitive applications, traffic from continuous polling and the resultant data transfers can easily impact performance. These problems can be addressed by using IMAs. The fault detection of an IMA can be pre-defined and the IMA is dispatched from the NMS and travels from node to node. The IMA can monitor, capture and process packets locally.

The intelligent mobile agents-based architecture designed in this paper is shown in Fig.2. The architecture is comprised of a manger and each network element (NE) which runs a Mobile Agent Daemon (MAD, here we use aglet from IBM) within a Java Virtual Machine (JVM). The manager has a static agent(Ad: Agent determination) to detect fault and an mobile agent(MAx: Mobile Agent exchanging) to exchange information with all of intelligent mobile agent(IMA) in each NE using KQML. Each NE has an IMA which monitors ICMP packets and detects faults locally. The IMAs improve network fault detection ability by dynamic distribution of ICMP packet monitoring intelligence to the networked devices. Instead of querying each node in a client/server mode to gather the

information needed to perform fault detection, the IMA processes the fault detection locally, draws a simple fault and isolates symptoms from the observed ICMP packets. The MAx's goal is to visit all of the nodes that contain any information relevant to the fault detection and exchange information with all of IMAs.

The process of performing fault detection by an IMA consists of several steps:

Step 1. the manager equips an IMA with the ICMP packet monitoring intelligence required to accomplish the task and send it to each NE(steps are shown in Fig.2).

Step 2. the IMA in each NE isolates symptoms from observed ICMP destination unreachable (ICMP-DUR) packets locally.

Step 3. drawing simple faults from diagnosing the above symptoms.

Step 4. to suppress known symptoms and filter ICMP packets, the IMA must know the faults detected. So the simple faults detected locally and others in manager collected(step 7) from all the nodes should be sent to IMA, and the local simple faults should be transmitted to manager to be informed all the other IMAs.

Step 5. sending the isolated symptoms to the manger to be used to diagnose and detect fault.

Step 6. the manager diagnoses all the symptoms and detects faults to be sent to each IMA.

Step 7. the manager creates a mobile agent(MAx) which migrates to the nodes that need the results to exchange information with it using KQML.

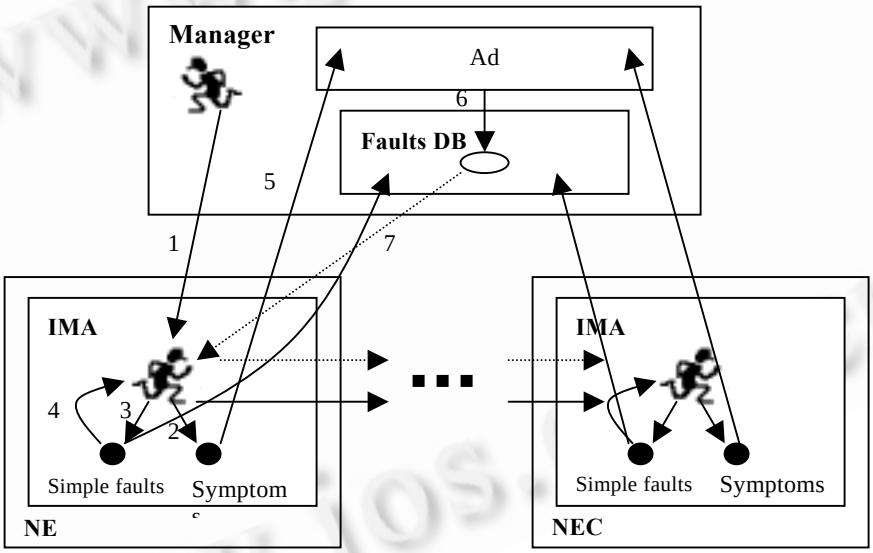

Fig.2 The intelligent mobile agents-based architecture

2.1 Structure of the intelligent mobile agent

In an IMA for fault detection there are essentially two parts: Packet Monitor and Symptom Isolation. The structure of the IMA is shown in Fig.3.

In a large-scale network, one fault is likely to generate a large quantity of ICMP packets. Checking all ICMP packets is very inefficient. If related ICMP packets are aggregated, so the burden of the management will decrease. Here we use symptom-based aggregation model, we focused on the source address and destination address of packets. Because the packets are essentially characterized by the sender and the receiver, and the information of which host sent or received packets is very important in fault detection, aggregation should primarily be based on these addresses.

The Packet Monitor captures ICMP packets and divided them, beforehand, into three categories based on the ICMP error type (Echo Request, Echo Reply, Destination Unreachable) and monitored individual categories. These three types of errors are observed more than other ICMP types. Then, the Packet Monitor collects all the relevant pieces of information (ICMP's error type, error code, original packet header which is included in the ICMP payload, etc.) by decoding all the aggregated ICMP packets, and sorts and separates the different Destination-IP addresses of unreachable destination (in the IP-header that is sent as data in the ICMP packet).

Each unreachable destination is a symptom which represents the unreachability of a different host. The symptoms are then arranged in frequency of occurrence and organized a "symptom table". The Symptom Isolation uses the symptom table to isolate symptoms and examine whether it was possible to guess the fault or not. If possible, then draw the simple faults. Filters are made corresponding to these simple faults and are used as suppression filters in Packet Monitor. At the same time, the Symptom Isolate sends these simple faults to faults DB and sends some of other symptoms to Ad in manager to be further processed.

To confirm a fault detected, the IMA can make NE create corresponding ICMP packets, e.g. when a unreachable host is detected, the IMA generally launch the "ping" command to the host to get some ICMP-DUR packets for confirming the host's existence.

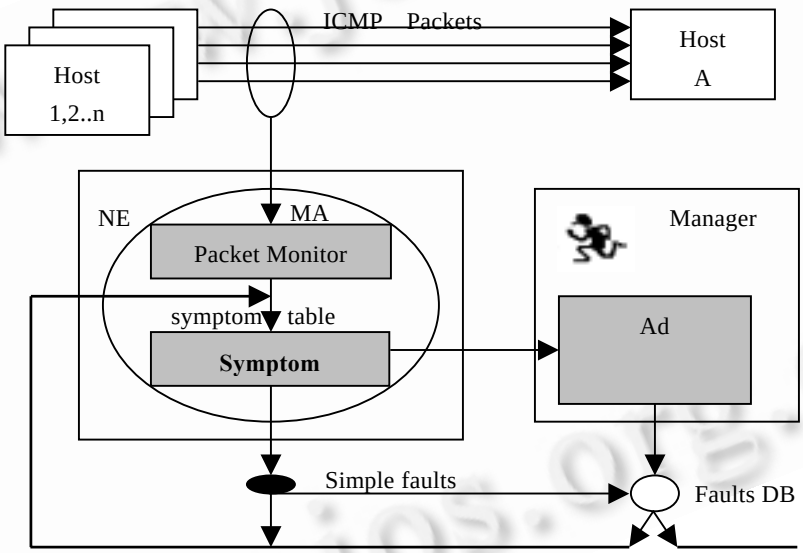

Fig.3 The structure of the intelligent mobile agent

2.2 Packet monitor

The structure of Packet Monitor module is shown in Fig.4. On being notified of an event due to the ICMP-Host-Unreachable packets the IMA would capture all ICMP-DUR packets and decode them to separate ICMP error type, error code and destination unreachable IP-addresses into the address table sorted by IP-address. Then, the filter is applied to suppress known symptoms and filter some packets to reduce burden of management. Filters are made corresponding to these symptoms and are used as suppression filters in the event detection phase. E.g. if the fault of host "192.9.200.1" being unreachable is detected, so the IMA maybe ignore some of the destination "192.9.200.1" unreachable ICMP packets during a period.

Here, we focus on ICMP echo request, echo reply, and destination unreachable

ICMP echo request and echo reply

An echo request can be used to decide whether a destination is reachable and any IP machine receiving an echo request is supposed to respond with an echo reply. An echo request is known as a 'ping'. Attackers generally launch the ping to network hosts for confirming the hosts' existence. This is known as a 'host scan' and most of the latest intrusions use this scan[15]. If a host scan has occurred, ICMP echo request packets flow from a specific source address to several destination addresses or ICMP echo reply packets flow from several source addresses to the specific destination address.

These attacks using ICMP packets are effectively aggregated to one symptom. If an address number is observed sequentially by aggregated packets, there is a high possibility that host scan has occurred. This result is of great help in preventing network breaches because network scan is a widely used tool by intruders to explore vulnerabilities in nodes or networks, and once a scan is detected, measures can be taken against intrusion.

ICMP destination unreachable

If, according to the information in the gateway's routing tables, the network specified in the Internet destination field of a datagram is unreachable, e.g. the distance to the network is infinite, the gateway may send a destination unreachable message to the Internet source host of the datagram. In addition, in some networks, the gateway may be able to determine if the Internet destination host is unreachable. Gateways in these networks may send destination unreachable messages to the source host when the destination host is unreachable. If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a destination unreachable message to the source host.

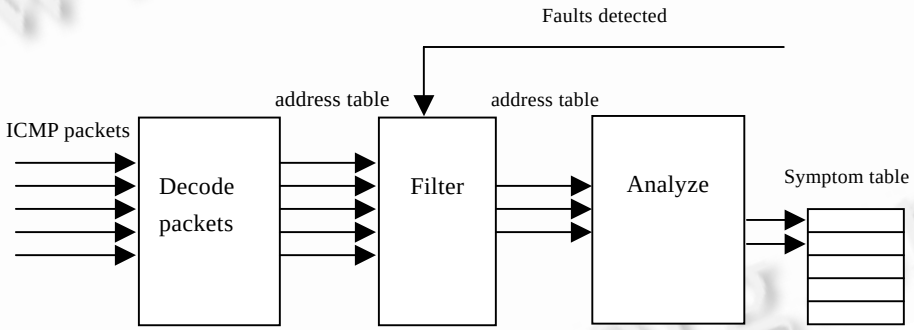

Fig.4 Packet monitor

2.3 Symptom isolation

The component of the Symptom Isolation module is shown in Fig.5. First, an event is detected by monitoring ICMP packets. Next, symptoms that triggered the detected event are analyzed. Significant symptoms are isolated from the aggregated characteristics of the subsequent cycle, in the last component.

The module is comprised of two parts. An event, E, is detected in the Event-detection phase. This event indicates the presence of one or more symptoms of faults. The IMA focuses on the set of candidate symptoms(S'), $S' \subseteq S$, S={monitored symptoms}, the candidate symptoms have triggered the event E. The IMA then isolates significant symptoms from the candidate symptoms and carries out the procedure for the corresponding faults. Detailed explanation of each phase follows.

Event detection

An event E is triggered when, the number, $n(s_i)$, of a candidate symptom s_i exceeds some threshold t_i for s_i. $n(s_i) > t_i \Rightarrow E = True$. However, for practical purposes, instead of applying the threshold to individual symptom s_i, it is more effective to apply the threshold to an set of monitored symptoms S, $\sum_i n(s_i) > T, s_i \in S \Rightarrow E = true$, the threshold T is in general set for a single symptom. It is clear that two or more symptoms may cooperate to trigger an

event. Thus the IMA has to carry out the process of identifying the symptoms that did trigger the event.

Symptom isolation

Though several symptoms cooperate to trigger an event, all the candidate symptoms are not significant, the IMA will separate and isolate the significant symptoms that is easy used to detect faults. The set of significant symptoms from the Symptom Isolation phase in the t^{th} cycle are suppressed in the subsequent event detection phase. The set of all the other significant symptoms will be transmitted to manager to deal with further analysis.

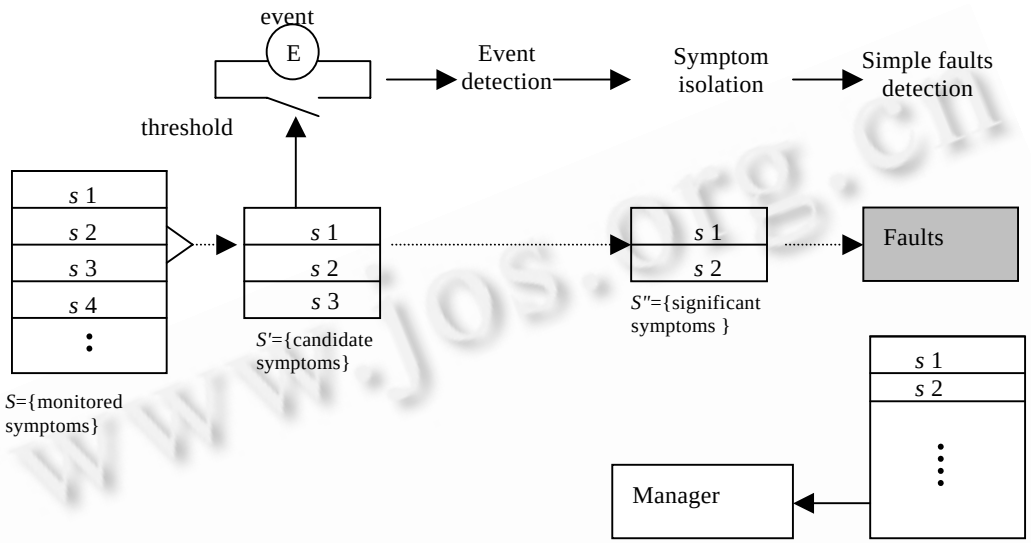

Fig.4 Symptom isolation

In this phase the IMA picks the significant symptoms *S*'' from *S*', using some criteria, for diagnosis. The IMA 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 sever 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 greater than the threshold are selected as significant symptoms indicating real faults, needing diagnosis.

3 Implementation

To evaluate the proposed architecture, we experimented on fault detection by monitoring the ICMP traffic in the network. The experimentation and system evaluation was carried out on our LAN network which is an Ethernet includes 35 nodes. We concentrated on the ICMP destination unreachable packets. We use the mobile agent system aglet 1.1b3[12] from IBM running on Java 1.1.8 from Sun. All the system is implemented using java technology running on the mobile agent aglet 1.1b3, includes ICMP packet monitoring, symptom isolation of IMA in each node, and MAx, Ad in manager.

In the experiment, we observed ICMP packets and divided them into three categories based on the ICMP error type (Echo request, Echo Reply, Destination Unreachable) and monitored individual categories. We set packet count thresholds corresponding to each type of ICMP packet. If the ICMP packet count per 5 minutes exceeded the threshold, ICMP packet capture is started. The duration of the capture must be long enough to ensure enough ICMP

packets are captured to make the symptom examination meaningful. But, in practice, the observed packet rate will vary widely with the type of fault. This makes it difficult to fix the capture duration. In our experiment, the traffic was sampled every 2 minutes for duration of 1 minute. For traffic analysis the sampled traffic of the most recent 360 minutes was used.

A part of results are shown in Figs.6~Fig.9. Figure 6 shows the total number of ICMP-DUR packets captured and the count of ICMP-DUR packets after the symptoms have been isolated at the same time in an hour. We can see that the number of packets further analyzed is less than that of packets captured and don't need to further analyze all the packets. This graph also shows the number of unresolved symptoms of faults present in the traffic is well within control. In Fig.7, we show the characteristics of some of the isolated symptoms corresponding to an hour period in Fig.6.

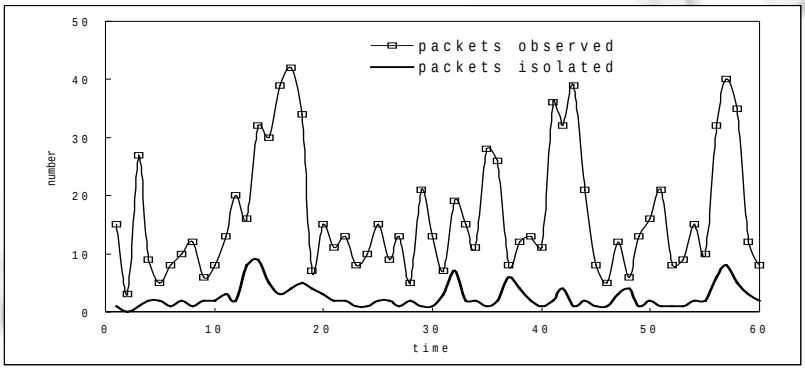

Fig.6 ICMP-DUR packets observed vs isolated

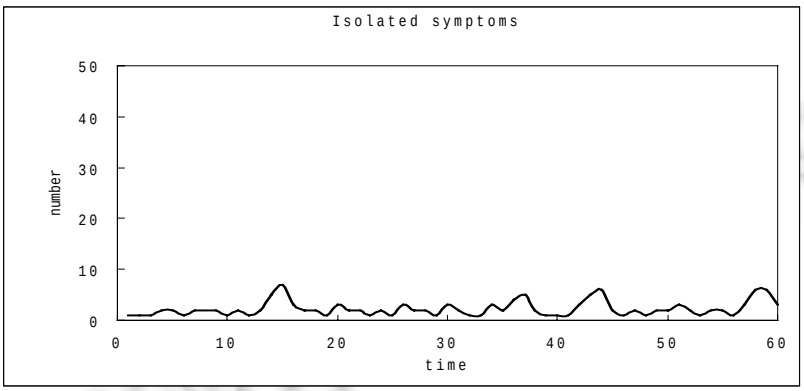

Fig.7 Isolated symptoms

To measure the impact on symptom detection capability we compared the performance of our system with a traditional system using RMON-Agents and packet monitoring. Both systems 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 threshold 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 one time using the new system, is 155 and 63, otherwise; and the number detected more than 8 times using the new system, is 27 and, 6 otherwise. It is evident that the symptom detection capability using the new approach based on intelligent mobile agents and packet monitoring is significantly improved.

The comparison of NMS traffic is shown in Fig.9. Experiment results show the NMS traffic in new system

based on IMAs is more less than that in traditional system. Table 3 shows recognized faults.

In a word, the experiment results show that using the intelligent mobile agents-based architecture for fault detection is more effective. When compared to classic client-server solutions, the architecture reduces network traffic and increases scalability, flexibility and robustness.

Table 3 Recognized faults

Fault	Count
Host unreachable	2
Protocol unreachable	1
Port unreachable	12
Prohibited by filtering	2

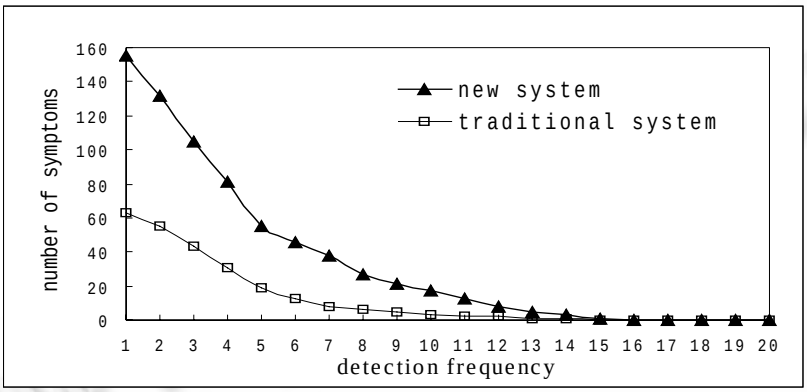

Fig.8 Comparison of number of symptoms detected

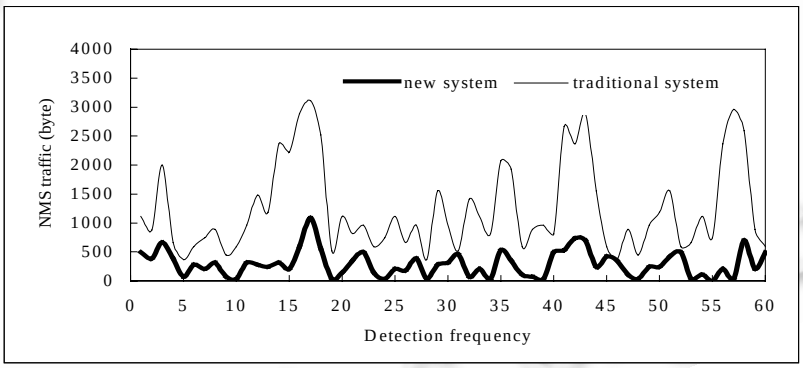

Fig.9 Comparison of NMS traffic

4 Conclusions

In this paper, we present an intelligent mobile agents-based architecture fault detection and have implemented the system using mobile agent system aglet and java language. In each IMA, we have focused on the issue of network fault detection by ICMP packets monitoring technique. The implementation of this architecture has demonstrated that intelligent mobile agents can be effectively used to find faults in a network context. The IMA performs simple fault detection and correlated with manager. It autonomously diagnoses the alarms and determines the cause of faults using a built-in network management intelligence.

An IMA has the capability to extract necessary data from the network element over a high bandwidth, local communication session. This does not consume overall network resources reducing the overall communication traffic. In addition, the IMA has the ability to integrate knowledge from both the manager and the network element. It performs inference asynchronously at network elements allowing the manager to focus on other tasks. So, The proposed architecture is very scalable to large networks.

References:

[1] LaBarre, L. Management by exception: OSI event generation, reporting and login. In: Proceedings of the 2nd International Symposium on Integrated Network Management. 1991.

[2] Kliger, S., Yemini, S., Yemini, Y., *et al*. A coding approach to event correlation. In: Proceedings of the 4th International Symposium on Integrated Network Management. IFIP, 1995. 266~277.

[3] Mansfield, G., Ouchi, M., Jayanthi, K., *et al*. Techniques for automated network map generation using SNMP. In: Proceedings of the INFOCOM'96. 1996. 473~480.

[4] Akira, K., Kohei, O., Nei, K., *et al*. A simple packet aggregation technique for fault detection. International Journal of Network Management, 2000,10:215~228.

[5] Postel, S.J. Internet Control Message Protocol. RFC0792, 1981.

[6] Ohta, K., Mori, T., Kato, N, *et al*. Divide and conquer technique for network fault management. In: Proceedings of the 5th International Symposium on Integrated Network Management. IFIP, 1997. 675~678.

[7] Mori, T., Ohta, K., Kato, N., *et al*. The Dynamic symptom isolation algorithm for network fault management and its evaluation. IEICE Transactions on Communications, 1998,E81-B(12):2471~2480.

[8] Kanamaru, A., Ohta, K., Kato, N., *et al*. Simple aggregation technique for fault detection based on packet monitoring. In: Proceedings of the Symposium on Performance Evaluation of Computer and Telecommunication Systems. 1999. 395~399.

[9] Stallings, W. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Reading, MA: Addison-Wesley, 1999.

[10] Perkins, D.T. RMON remote monitoring of SNMP-Managed LANs. Englewood Cliffs, NJ: Prentice Hall, 1999.

[11] White, T., Bieszczad, A., Pagurek, B. Mobile agents for network management. IEEE Communications on Surveys, 1998,1(1):2~9.

[12] http://www.trl.ibm.co.jp/aglets/.

[13] http://agents.umbc.edu/.

[14] Postel, S.J. Requirements for IP Version 4 Routers. RFC1812, 1995.

[15] Mansfield, G., Ohta, K., Takei, Y., *et al*. Towards trapping wily intruders in the large. In: Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection. 1999.

一种基于智能移动代理的网络故障检测系统

张普含, 孙玉芳

(中国科学院 软件研究所,北京 100080)

摘要: 随着网络规模的急剧扩大和结构的日趋复杂化,网络故障管理越来越重要.在一个复杂的通信网络中,故障是不可避免的,但是对故障的及时探测和识别对于提高网络的可靠性是非常重要的.监测数据包是网络故障检测中常用的方法,但是在大规模网络中会产生巨量的数据包.为此,人们提出了几种方法.但是这些方法都是建立在集中式管理体系结构之上的,因而没有良好的可扩展性、灵活性和本地处理能力.针对这些问题,提出了一种基于智能移动代理的网络故障检测系统结构,并用 Java 和 aglet 实现.实验表明,这种系统结构对于网络故障的检测是非常有效的.

关键词: 网络故障探测;智能移动代理;ICMP 数据包监测;网络管理

中图法分类号: TP393 文献标识码: A