Autonomous Network Monitoring System based on Agent-mediated Network Information

Sameera Abar†, Susumu Konno† and Tetsuo Kinoshita†,

Tohoku University, Sendai, Miyagi, JAPAN

Summary
The growing complexity of communication networks and their associated information overhead have made network management considerably difficult. This paper presents a novel Network Management Scheme based on the novel concept of Active Information Resources (AIRs). Many types of information are distributed in the complex network, and they are changed dynamically. Under the AIR scheme, each piece of information in a network is activated as an intelligent agent: an I-AIR. An I-AIR has knowledge and functionality related to its information. The I-AIRs autonomously detect run-time operational obstacles occurring in the network system and specify the failures' causes to the network administrator with their cooperation. Thereby, some network management tasks are supported. The proposed prototype system (AIR-NMS) was implemented. Experimental results indicate that it markedly reduces the network administrator workload, compared to conventional network management methods.

Key words:
Active Information Resource, Network Monitoring, Intelligent Agent, Multi Agent System, Expert System.

1. Introduction

In recent years, computer communication networks have grown dramatically both in size and complexity. Moreover, they comprise heterogeneous multi-vendor environments. Traditionally, network management activities have been performed by network managers. However, these activities are becoming more demanding and data-intensive because of the rapid growth of modern networks. For those reasons, automation of network management activities has become necessary. For managing these huge distributed network systems, manual procedures have become tedious.

A typical approach to network management is centralized, static, polling-based management that involves high-capacity computing resources at the centralized platform including commercially available management tools. As managed components become more numerous, the amount of network traffic, which should be managed, have increased accordingly. Consequently, in centralized management systems, the management traffic might eventually oppress the network bandwidth. Even where the management platform uses several distributed management stations, the huge bulk of management traffic remains concentrated around those stations [1]. The overwhelming volume and complexity of the information involved in network management imparts a terrible load [2].

Furthermore, in view of the dynamic nature of evolving networks, future network management solutions need to be flexible, adaptable, and intelligent without increasing the burden on network resources. The rapid of network systems has posed the issues of flexibility, scalability, and interoperability for the centralized paradigm. Even though failures in large communication networks are unavoidable, quick detection and identification of the causes of failure can fortify these systems, making them more robust, with more reliable operations, thereby ultimately increasing the level of confidence in the services they provide [3]. Motivated by these considerations, the proposed approach is intended to provide an intelligent, adaptive and autonomous network monitoring support paradigm for communication network systems.

A network monitoring support method based on the activated information is proposed in this paper. In this method, the distributed information in a computer network is activated using the concept of Active Information Resource (AIR). In the AIR scheme, each unit of distributed information has knowledge and functionalities related to utilization of the information resource as well as its information. In our experiment network system, each activated information AIR (I--AIR) is developed as an intelligent agent. The proposed framework simplifies network monitoring for the administrator. Experiments were performed to investigate the effectiveness of the proposed method.

The remainder of the paper is organized as follows. Section 2 presents an overview of the AIR concept and conversion of the dynamic status information as I-AIRs. The detailed design and implementation considerations of I-AIRs in the proposed prototype system are discussed in Section 3. Experimental results, along with the system's performance evaluations are outlined in Section 4. Finally, the conclusions and future issues are presented in Section 5.

Manuscript received February 5, 2008
Manuscript revised February 20, 2008

2. Automated Network Monitoring based on Activated Network Information

For monitoring of communication network by an administrator, much status information distributed in a network is required, such as network traffic, conditions of service processes, and application server logs. Commonly, the information is static; furthermore, an administrator must investigate them one by one, which places a necessary physical and mental load on the administrator.

In this study, therefore, this static information is activated to reduce the administrator's workload. For activation of status information, a concept of an active information resource (AIR) [4][5][6] is employed. Each unit of status information is wrapped as an AIR for activation; it is called I-AIR. An I-AIR has its original information resources along with related knowledge and functionalities. Several I-AIRs can cooperate autonomously based on their status information and knowledge. Consequently, our scheme can reduce network management loads by presenting the dynamic status information of the network resources during automatic detection and specification of network failures.

2.1 AIR concept

An AIR is defined as the distributed information resource enhanced with its knowledge as well as functionality to facilitate its resources. Fig.1 shows a conceptual model of an AIR with its support knowledge and functionality. The knowledge of an AIR typically consists of metadata of the information contents and their processing descriptions. The functionality of AIR is about how to analyze and process the users' query as well as defining the cooperation strategy among the multiple AIRs.

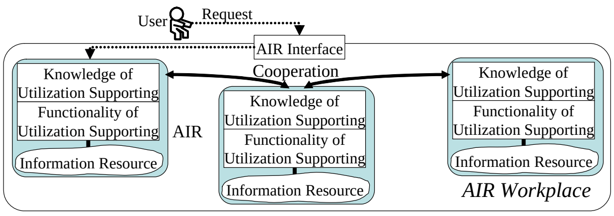

Fig.1 Active Information Resource

An AIR can be implemented using the multi-agent-based approach. Agent-based computing is known as a complementary way to manage the resources of distributed systems because of the increased flexibility in adapting to the dynamically changing requirements of such systems [7],[8].

Essential features of AIRs include:

- To extract and process the information contents in response to the query from user (or another AIR) in a knowledge-based manner.
- To interact actively and mutually to make full use of the information contents, the embedded support knowledge, and functionality.

The effectiveness of AIR has been employed in the context of diverse web-based information retrieval techniques. The prototype systems have exhibited very promising results.

2.2 Applying the AIR concept to Network Monitoring

Generally, the status information of the communication network is classifiable into two types: static information and dynamic information. For example, the relationship between IP addresses and Mac addresses, host names, domain names, IP-routing, etc., are included as static network information. On the other hand, the dynamic information includes number of packet traffic, RMON-MIB, SNMPv2-MIB, logs of network services, etc. To apply the concept of AIR to both types of information for network monitoring, each unit of information is converted to an AIR to form a so-called I-AIR.

Conventionally, an administrator collects various status information through periodical polling. An administrator aggregates the data and decides the status of the network system using his know-how. This task can be disaggregated into three stages, such as detection, recognition, and specification of the failure. This task requires much experience as a network manager; therefore, a beginner cannot be employed as an administrator.

To support the empirical task of the administrator, an I-AIR includes diverse knowledge and functionality in addition to its original data. For example:

- meta-knowledge about information resources
- knowledge about failure condition (threshold)
- knowledge about cooperation with another I-AIR
- functionality to handle original data

Using this additional knowledge and functionality, I-AIRs can mutually cooperate. The following tasks can be partially supported by AIR:

- distributed and effective monitoring of network system
- detection of network failure using a threshold
- processing of information resources according to the failure with its functionality
- improvement of reliability of detection, recognition, and specification of the failure through cooperation among AIRs

These features can reduce the overall workload of the administrator.

3. Design and Implementation of I-AIR

In this section, the design of an I-AIR is discussed. The design comprises three vital ingredients: internal support knowledge, functionality for sharing the information contents, and specifications of the information resource itself.

3.1 Design of Knowledge in I-AIR

The support knowledge for sharing information contents is the empirical knowledge of network management which inspects the status information of the network for occurring faults. Essential components of this knowledge are as follows:

I-AIR Identification Knowledge (ID): The ID includes an identification number, task number of I-AIR, etc.

Knowledge about Information Resource (IR): The IR includes a type, an update-time, a format type, etc.

Knowledge about Failure Inspection (FI): The FI includes two types of knowledge to inspect the failure: text information to be detected in logs, and a threshold of packets, etc.

Knowledge about Network Periodic Investigation - Control Method (CM): The CM includes the polling time and other conditions for updating of the information resource.

Knowledge about Cooperation Protocol (CP): The CP includes protocol sequences for cooperation with other AIRs.

The knowledge contained in an I-AIR as ID, IR, and CP is required mainly in order to operate on the information resource and facilitate communication and cooperation among the I-AIRs. The preeminent characteristic of I-AIR is its autonomous monitoring mechanism, which is supported via FI and CM for the inspection and investigation of the obstacles that hinder the normal network operation. Thus, the performance of I-AIRs in the proposed technique relies heavily on the design of various types of internal support knowledge. Fig.2 shows the I-AIR's knowledge representation scheme as BNF.

```
I-AIR ::= <ID> <IR> <FI> <CM> <CP>
 <ID> ::= <Air id> <Workplace id> <Task id>
              <Air id>          ::= <Value>
              <Workplace id>    ::= <Value> <Air id> | <Name> <Air id>
                                  | <Value> | <Name>
              <Task id>         ::= <Value>
              <Name>            ::= <Character>+
              <Value>           ::= <Number>+
              <Character>       ::= '[a-z A-Z]' | '.' | ':'
              <Number>          ::= '[0-9]'

 <IR> ::= <Info type> <Path> <Format type> <Time>
              <Info type>       ::= <Name>
              <Path>            ::= '/'<Name> | '/' <Name> <Path>
              <Format type>     ::= 'xml' | 'text'
              <Time>            ::= <Year> '/' <Month> '/' <Day> '/'
                                  <Hour> ':' <Minute> ':' <Second>
              <Year>            ::= <Value>
              <Day>             ::= <Value>
              <Hour>            ::= <Value>
              <Minute>          ::= <Value>
              <Second>          ::= <Value>

 <FI> ::= <Failure name> <Check name> <Check string> <Check info>
              <Failure name>    ::= <Name>
              <Check name>      ::= <Name>
              <Check string>    ::= <Name>+
              <Check info>      ::= <Exist>
                                  | <Threshold> <Relation>
                                  | <Threshold> <Threshold> <Relation>
              <Exist>           ::= 'yes' | 'no'
              <Threshold>       ::= <Value>+
              <Relation>        ::= 'over' | 'less' | 'between'

 <CM> ::= <Method name> <Arguments> <Trigger info>
              <Method name>     ::= <Name>
              <Arguments>       ::= <Argument>+
              <Trigger info>    ::= <Interval> | <Last check>
              <Argument>        ::= <Value> | <Name>
              <Interval>        ::= <Value>
              <Last check>      ::= <Name>+

 <CP> ::= <protocol>+
              <protocol>        = Cooperation protocol
```

Fig.2 Knowledge representation scheme of an I-AIR

3.1.1 I-AIR Identification Knowledge (ID)

<Air id> represents the Identification Knowledge of an AIR. <Workplace id> is the identifier for the environment where the I-AIRs are instantiated and cooperate actively. <Task id> is the identifier for the task. <Value> tag specifies the number of the task which is in the midst of processing. The <Character> and <Number> are optionally expressed.

3.1.2 Knowledge about Information Resource (IR)

<IR> characterizes the knowledge about the information contents. This knowledge is utilized when I-AIRs receives the request for performing some operation. <Info type> distinguishes among the information resources, for instance, Postfix and syslog etc. <Path> points the I-AIRs towards the path of the information resource, e.g., /var/log/maillog. <Format type> specifies the composition of contents as XML or text file. <Time> serves to indicate the instant (e.g., 2004/04/01/11:11) when the information resource is updated.

3.1.3 Knowledge about Failure Inspection (FI)

<FI> represents a part of experiential management knowledge of expressed as the internal state support knowledge of an I-AIR. This knowledge is expressed as the production rule instructions in an I-AIR. <FI> inevitability constitutes the core knowledge as it used for monitoring the operational state of the network for effectively detecting the anomalies. For inspecting a wide range of failures, two entities are particularly crucial to be taken into account in the design of <FI>:

- Error Information (character string)
- Packet quality with threshold value

<Failure name> indicates the name of the reported obstacle which obstructs the normal functioning of the network. With <Check name> I-AIR executes the investigation name, <Check string> aids in the inspection of an obstacle from the information resource with meta-information. For the <Check info> when the <Check name> shows the obstacle character string, <Exist> is described. For example, if the run-time “ping” command indicates an error, <Check name> reports the character string as “Request timed out”, <Exist> value indicates “yes” for the existence of the failure. If the system is behaving in a normal fashion, then the <Exist> value is “no” which indicates that the process name does not exist. Additionally, when the data traffic is observed from the packet log, <Threshold> value helps in the determination of fixing of the obstacle. <Relation> is described in concurrence with the threshold value. For example, if the packet flow is above or below the threshold, the obstacle is investigated.

The knowledge of <FI> is defined in such a manner that the supplementary information is provided to the character string for the inspection of failure. Hence, a part of the knowledge is modified dynamically in relation to the functionality for sharing the information contents. This has been incorporated successfully for the scanning of data traffic flow and sending/receiving mail scenarios.

3.1.4 Knowledge about Network Periodic Investigation – Control Method (CM)

<CM> deals with the knowledge for investigating the network failures repeatedly, and is incorporated vis-à-vis the functionality of I-AIRs for sharing the information contents. <Method name> refers to the command name for executing the functionality for sharing the information resource. <Arguments> and <Trigger Info> group describes the information which initiates the investigation process. More specifically, with <Trigger Info>, the knowledge is utilized during the course of automatic constant investigation for the network failures. <Interval> holds the time interval at which the information resource is updated. Similarly, for <Last check>, at the time of receiving the request, the investigation object which has already been executed is described by the message, and the investigation name from the pre-defined list is again initiated for the investigation job. For actualizing the investigation function, the command name describing some particular function to be executed is expressed through the <Method name>. Hence just a modification in the contents of <Method name> makes it possible to assimilate a diverse type of information. Thus with the control method, the information source is replenished dynamically, whereas the thorough monitoring of the failure object is done by <FI>.

3.1.5 Knowledge about Cooperation Protocol (CP)

<CP> represents the knowledge regarding the interaction / cooperation protocol amongst various I-AIR. For directing the autonomous cooperation mechanism, two types of protocols (inform-failure protocol, and report protocol) have been defined. Fig.3 depicts the I-AIRs cooperation strategy. During the course of autonomous monitoring of the network for failures by I-AIRs, the Inform-failure protocol broadcasts the failure information to each workplace. Hereafter, each I-AIRs upon receiving the failure information makes decision based on its support knowledge whether it is required or not to take further action for determining the cause of failure. Then the relevant I-AIRs explore the details about the failure and broadcast the result to other I-AIRs. Hence, the cooperation among the I-AIRs is facilitated until the root cause of failure is identified. With report protocol, the exact information about the obstacle cause is forwarded to AIR-interface.

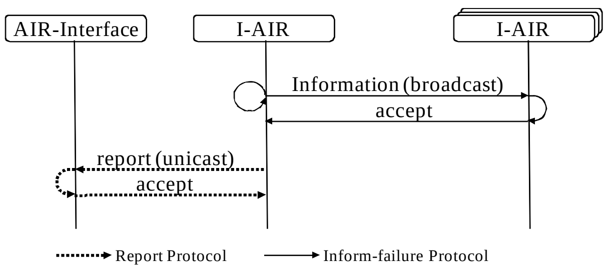

Fig.3 Protocol sequence in cooperation

3.2 Design of Functionality of I-AIR

I-AIRs' functionality deals with the sharing and processing of the information resource for cooperative problem solving during the active fault monitoring and detection phases. In this regard, the design of some essential features is crucial as follows:

- Functionality as an Interface to I-AIR internal support knowledge

- Functionality for processing the information resource
- Functionality for transmitting the processed results to other I-AIRs
- Functionality for inspecting the obstacle with respect to the pre-defined threshold

3.3 Design of Information Resource

Two I-AIR information resource types are described here.

- Simple text format
- RDF/XML syntax specification

The RDF/XML language is a W3C-recommended framework for describing information resources using machine-readable metadata, which brings about an unprecedented level of automation for the representation and retrieval of information. The plain-text format consists of log-information that is acquired through the Syslog (a standard logging solution on UNIX and Linux systems). In the proposed approach, the I-AIR functionality extracts a diverse type of log-information during operational management scenarios and converts it to RDF/XML format specifications.

3.4 State Transition Diagram of I-AIR

The state transition diagram of I-AIR is shown in Fig.4 respectively. An I-AIR has eight states. Each state is described below.

Wait: *Wait* is initial condition of I-AIR. If I-AIR received investigation requirement, state will transit to *Judge Investigate Requirement*. If information is updated, state will transit to *Verify Information*. If I-AIR received detail information requirement, state will transit to *Create and Send Investigation Request*. If I-AIR detect Detail flag is on, state will transit to *Create Detail Information*.

Judge Investigate Requirement: I-AIR judges whether requirement is possible or impossible. The impossible case, state will transit to *Wait*. When possible, state will transit to *Investigate Information*.

Investigate Information: I-AIR investigates own information. If I-AIR succeeds the investigation, state will transit to *Forward Result*.

Forward Result: I-AIR will send investigating information to other AIR, and state will transit to *Wait*.

Verify Information: If failure information is included in status information, state will transit to *Create and Send Investigation Request*. If it is not included, state will transit to *Wait*.

Create and Send Investigation Request: I-AIR will create and send investigation request to other AIR, and state will transit to *Wait*.

Create Detail Information: When Detail flag is on, I-AIR constructs a detail information, and its forward to other AIR. If it is succeeded, state will transit to *Wait*.

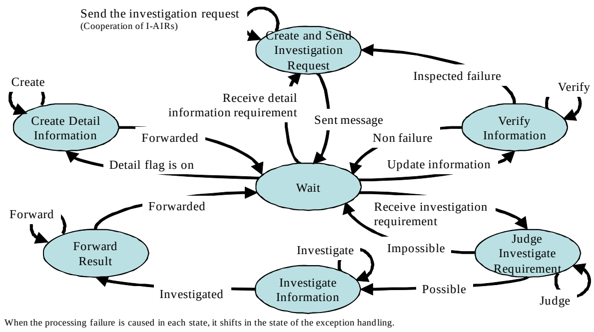

Fig.4 State Transition Diagram of I-AIR

3.5 Implementation of I-AIR

A multi-agent-based approach was adopted for implementation of I-AIRs in the proposed technique. For the effective realization of I-AIR support knowledge and functionality, the multi-agent system is a highly pragmatic choice. The I-AIRs realized with the software agents render the I-AIRs active, which, after being invoked by an outside event, can autonomously perform the task of cooperative problem-solving. The proposed system architecture is supported by an Agent-based Distributed Information Processing System (ADIPS) framework [9], which is a flexible computing environment for designing multi-agent systems. Table 1 illustrates the I-AIRs developed in this study. Fig.5 showed the example of describing the knowledge of I-AIR (No.15).

Table 1: Examples of implemented I-AIRs for network monitoring

I-AIR No.	Function	I-AIR No.	Function
1	Network Disconnection detector	11	DNS server process checker
2	NIC configuration failure detector	12	SMTP server process checker
3	SPAM mail detector	13	POP server process checker
4	MSBlaster attack detector	14	DNS connection checker
5	Mail send/receive error detector	15	Network route to host checker
6	TCP/IP stack failure checker	16	Kernel information checker
7	NIC configuration failure checker	17	Lease IP address checker
8	HUB failure checker	18	Mail server error checker
9	Router failure checker	19	Number of SPAM mail
10	Communication failure checker		

4. Evaluate the Effectiveness of I-AIR in Actual Monitoring Task

To evaluate the prototype system's effectiveness, an experimental NMS system, called AIR-NMS, was set up in the laboratory. The network administrator performs the management task according to the conventional manual method, as well as with the I-AIRs based proposed system. He also measures the performance of the proposed approach adopted for the automation of network functions. In the experiment, the time and the number of procedures executed to correct the obstacle were measured after a network obstacle was reported to a subject.

(ID	:air id	"i-air@w1:pcB1.example.com"
	:workplace id	"w1:pcB1.example.com"
	:task id	"0123456789")
(IR	:info type	"ping_result"
	:path	"stdout"
	:format type	"text"
	:time	"2007/01/01/00:11:22")
(FI	:failure name	"NIC problem"
	:check name	"Ping_NIC_pcB"
	:check string	("100% packet loss"
		"Destination net unreachable"
		"TTL expired in transit"
		"Ping request could not find"
		"unknown host")
	:check info	(CI :exist "yes"))
(CM	:method name	"ping"
	:arguments	("-c"
		"4"
		"172.17.1.2")
	:trigger info	(TI :interval "60000"))
(CP	:protocol	"Inform-failure Protocol"
		"Report Protocol")

Fig.5 Example of describing knowledge of I-AIR (No.15)

4.1 Experimental Network

Fig.6 demonstrates the practical setup of the environment for experimenting with the I-AIRs. The network system comprises a 100BASE-TX Ethernet with a firewall configured as a Network Address Translation (NAT) firewall, a router, and various personal computers (PCs) arranged in four subnetworks. Subnetwork A is configured as a Demilitarized Zone range 172.16.0.0/24. The server (sevA1) DNS and Mail application settings are configured. The other three subnetworks (B, C, D) have IP-addresses in the order given as 172.17.1.0/24, 172.17.2.0/24, and 172.17.3.0/24. Moreover, the network management console for managing the whole setup resides in pcB1 of subnetwork B. In subnetwork C, there is a desktop-type PC system (pcC1) with a fixed IP address from the DNS server, and a notebook computer (pine) which acquires the IP-addresses through the DHCP. In addition, Fig.4 depicts the nodes (PCs, routers, firewall etc.) of the experimental network system. Each node shows the corresponding AIR workplace where the I-AIRs operate actively. For each node, about 15 AIRs were implemented. This implies that nearly 140 I-AIRs were incorporated within the experimental setup. A Linux operating system was used in each PC.

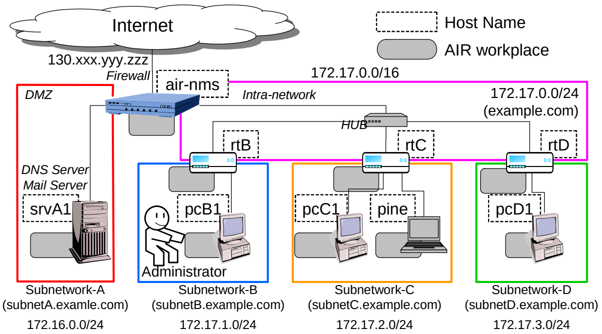

Fig.6 Construction of Network System and AIR-NMS

4.2 Experimental Methodology

Two kind of experimental methods have been designed, and for each method, five persons having expertise of managing computer communication systems have been employed. Two tactical measures are adopted for determining the effectiveness of the proposed prototype system has been taken into account:

1. Monitoring the network with the OS-default network management tools: Several failures obstructing the normal operation of network system are generated and accordingly it is required to restore the network services manually with the client management tools. Also, the time elapsed between the notification of failure to its remedy is measured.

2. Monitoring the network utilizing the I-AIRs: The obstacles are detected by the communication / cooperation mechanism of I-AIRs which are then reported to the I-AIR interface, then it is required to rectify the occurring failures. In this case also the time is measured from the point when the obstacle information is presented on the interface to the absolute restoration.

Hence, after some network obstacle has been reported and then corrected, the time is measured as well as the number of procedures executed to restore the network to its normal operation. These criteria serve as the index for measuring the practical worth and applicability of the automated I-AIR notion thereby determining the extent to which the burden of the network administrator has been reduced.

4.3 Experiment I: Various Application Scenarios

In this experimentation technique, several obstacle circumstances are generated and then inspected with and without I-AIR based system. These obstacles might occur by various causes. The task of a subject is to determine only one cause of a failure.

1 Cannot Connect to the Specific Host: In this case, file-transfer from pcD1 to pcB1 is not possible. A

rare cause has been presumed, that is, a problem with the settings of Network Interface Card (NIC) of the host computer (pcB1).

2 Transmission of Spam Mail: In this case, a spam mail is transmitted from pcD1. However, the originating location of spam is concealed, so it is required to detect accurately the host that sends out the illicit message.

3 Slow Network: This delinquency is reported in the case of accessing World Wide Web (WWW) connection. The notebook PC (pine) was infected through an attack (from MSBlaster from outside source) at the port 135, thereby hindering its access to the Internet.

4 Mail Sending/Receiving Error: Here, the client network encounters the problem in sending/receiving email because the reason that the SMTP server process is down.

Table 2: Experimental results (Exp.1)

1. Cannot Connect to the Specific Host

	A		B		C		D		E		Average	
	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step
no I-AIR	1056	20	756	20	680	22	771	20	282	40	709.0	24.4
I-AIR	99	5	51	2	125	4	226	5	52	2	110.6	3.6
I-AIR / no I-AIR (%)	9.4	25.0	6.7	10.0	18.4	18.2	29.3	25.0	18.4	5.0	15.6	14.8

2. Transmission of SPAM Mail

	A		B		C		D		E		Average	
	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step
no I-AIR	1096	24	221	4	901	23	1155	26	92	5	693.0	16.4
I-AIR	49	3	93	3	83	4	129	2	40	2	78.8	2.8
I-AIR / no I-AIR (%)	4.5	12.5	42.1	75.0	9.2	17.4	11.2	7.7	43.5	40.0	11.4	17.1

3. Slow Network

	A		B		C		D		E		Average	
	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step
no I-AIR	208	3	205	3	330	9	323	3	682	35	349.6	10.6
I-AIR	528	4	53	1	61	1	63	1	94	1	159.8	1.6
I-AIR / no I-AIR (%)	253.8	133.3	25.9	33.3	18.5	11.1	19.5	33.3	13.8	2.9	45.7	15.1

4. Mail Sending/Receiving Error

	A		B		C		D		E		Average	
	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step
no I-AIR	996	31	369	16	680	22	565	7	1499	49	821.8	25.0
I-AIR	98	4	59	2	125	4	81	2	73	2	87.2	2.8
I-AIR / no I-AIR (%)	9.8	12.9	16.0	12.5	18.4	18.2	14.3	28.6	4.9	4.1	10.6	11.2

Management experience: A. 1year, B. 2year, C. 2year, D. 3year, E. 7year

Results: The experimental results were compiled into Table 2. The results show that, for each failure situation, with the inclusion of I-AIRs, the management load related to the time taken to resolve a certain fault as well as the number of steps necessary to locate the cause of failure was reduced to an average 20%.

Table 3: Assumed failure causes: Mail Sending / Receiving Error (Exp.2)

Problem	Causes
Cable problem	a. Cable was disconnected.
Port problem	b. The 25th port was closed.
	c. The 110th port was closed.
DNS Server problem	d. DNS Server process was downed.
	e. Configration was not available.
Mail Server problem	f. Mail Server process was downed.

Table 4: Experimental results among individual administrators (Exp.2)

	F			G			H			I			J		
		Time	Step		Time	Step		Time	Step		Time	Step		Time	Step
no I-AIR	d	158	9	b	566	8	e	929	23	f	235	5	a	655	19
	e	743	24	d	871	12	b	339	9	c	615	9	f	182	5
I-AIR	a	51	1	f	104	2	c	82	3	a	40	1	b	86	2
	f	85	4	c	106	2	d	52	3	e	74	2	e	128	6
I-AIR / no I-AIR (%)	\	15.1	15.2	\	14.6	20.0	\	10.6	18.8	\	13.4	21.4	\	25.6	33.3

Management experience: F. 1year, G. 2year, H. 2year, I. 3year, J. 7year

Table 5: Experimental results among individual failures (Exp.2)

	a		b		c		d		e		f	
	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step	Time	Step
no I-AIR	655	19	566	8	615	9	158	9	743	24	235	5
	-	-	339	9	-	-	871	12	929	23	182	5
I-AIR	51	1	86	2	106	2	52	3	74	2	85	4
	40	1	-	-	82	3	-	-	128	6	104	2
I-AIR / no I-AIR (%)	6.9	5.3	19.0	23.5	15.3	27.8	10.1	28.6	12.1	17.0	45.3	60.0

4.4 Experiment II: One obstacle from various causes

An application scenario is tested against various causes for the occurrence of a specific failure condition to demonstrate the flexibility of the proposed approach using I-AIRs. Furthermore, these causes do not occur necessarily in any fixed pattern. The checks to detect these causes are performed randomly. However, using I-AIRs is advantageous because every check is done only once during the course of the fault-localizing process. The failure cause is detected and the main cause behind the failure is reported to the network operator actively.

Table 3 depicts the failure situation "Mail Sending / Receiving Error" with some possible causes underlying the occurrence of this anomaly. The task of the subject is to determine the cause of this error.

Results: Experimental results computed by each manager while resolving the mail sending / receiving anomaly were compiled into Table 4. Additionally, the results corresponding to each failure cause were accumulated into Table 5. The results demonstrate that the network management overhead regarding the time taken to resolve a certain fault, along with the number of steps necessary to locate the cause of failure, were reduced to 20% on average, which concurs exactly with the results of Experiment 1.

5. Summary

This paper presented a novel technique for the automation of management tasks for communication network systems. The foundation of the proposed framework is the use of I-AIRs, which, through active mutual interaction and with the functional network system, can resolve various network-failure situations. A part of the I-AIR knowledge is modified dynamically and frequently according to the operational characteristics of the network. Moreover, experimental results demonstrated a marked reduction in the administrator workload through the use of the proposed automated network monitoring and fault detection functions.

References

[1] Stephan, R., Ray, P., Paramesh, N.: Network management platform based on mobile agent. International Journal of Network Management 14 (2003) 59–73

[2] Consens, M., Hasan, M.: Supporting network management through declaratively specified data visualizations. In: IEEE/IFIP 3rd International Symposium on Integrated Network Management. (1993) 725–738

[3] Bouloutas, A., Calo, S., Finkel, A.: Alarm correlation and fault identification in communication networks. IEEE Transactions on Communications 42(2,3,4) (1994) 523–5334.

[4] Kinoshita, T.: A method for utilizing distributed information resources effectively: Design of active information resource (in japanese). In: Technical Report of IEICE (Japan) AI99-54. (1999) 13–19

[5] Li, B., Abe, T., Sugawara, K., Kinoshita, T.: Active information resource: Design concept and example. In: The 17th International Conference on Advanced Information Networking and Applications. (2003) 274–277

[6] Li, B., Abe, T., Kinoshita, T.: Design of agent-based active information resource. In: The 1st International Conference on Agent-Based Technologies and Systems. (2003) 233–244

[7] Sameera Abar, Hideaki Hatori, Toru Abe, Tetsuo Kinoshita, "A Semantic Knowledge Model for Agent-based Network Management System", A.Abraham et al, (eds.) Soft Computing as Transdisciplinary Science and Technology (Proc. 4th IEEE Int. Workshop WSTST05), Advanced in Computing, Springer, pp.808-818, 2005.5.

[8] Papavassiliou, S., Puliafito, A., Tomarchio, O., Ye, J.: Mobile agent-based approach for efficient network management and resource allocation: Framework and applications. IEEE Journal on Selected Areas in Communication 20(4) (2002) 858–872

[9] Fujita, S., Hara, H., Sugawara, K., Kinoshita, T., Shiratori, N.: Agent-based design model of adaptive distributed systems. Journal of Applied Intelligence 9(1) (1998) 57–70

Sameera Abar is a Ph.D student of Graduate School of Information Science, Tohoku University. She received a M.S. degree from Quaid-i-Azam University, Islamabad, Pakistan, in 1997.

Susumu Konno is a senior assistant professor of Graduate School of Information Sciences, Tohoku University, Japan. He received his Ph.D degree in information engineering form Chiba Institute of Technology in 2002. His research interests include flexible networks, agent-based computing, knowledge engineering and symbiotic computing. He is a member of IEEE, AAAI, IEICE and IPSJ.

Tetsuo Kinoshita is a professor of Information Synergy Center and Graduate School of Information Science, Tohoku University, Japan. He received his Dr.Eng. degree in information engineering from Tohoku University in 1993. His research interests include agent engineering, knowledge engineering, knowledge-based systems and knowledge-based systems. He received the IPSJ Research Award, the IPSJ Best Paper Award and the IEICE Achievement Award in 1989, 1997 and 2001, respectively. Dr. Kinoshita is a member of IEEE, ACM, AAAI, IEICE, IPSJ, JSAI, and Society for Cognitive Science of Japan.