Design and Analysis of a Proactive Application Management System (PAMS)

Salim Hariri
HPDC Laboratory
The Department of Electrical and Computer Engineering, University of Arizona
Tucson, AZ 85721, USA
Hariri@ece.arizona.edu,
http://www.ece.arizona.edu/~hpdc

Yoonhee Kim
The Department of Electrical Engineering and Computer Science, Syracuse University
Syracuse, NY 13210,USA
yhkim@top.cis.syr.edu

Abstract

Management of large-scale Network-Centric Systems (NCS) and their applications is an extremely complex and challenging task due to factors such as centralized management architectures, lack of coordination and compatibility among heterogeneous network management systems, and the dynamic characteristics of networks and application bandwidth requirements, just to name a few. The goal of this research is to develop a hierarchical framework to achieve end-to-end intelligent proactive network management system that can be used to manage large-scale network centric systems and their applications. This framework will provide the ability to write management programs to manage any required function or property (performance, high assurance, fault, quality of service, etc.) of the network-centric systems and their applications during all the phases of their operations. Our ultimate objective is to consider the management of network-centric systems and applications starting from the design phase and forward rather than being after thought process. In this paper, we present a framework to develop proactive and adaptive management services and an implementation of a Proactive Application Management System (PAMS) based on that framework. Our implementation approach utilizes delegated mobile agents to implement the management functions required by any network-centric system and/or application. We will also present experimental results and evaluation of the management services offered by the PAMS prototype.

Keywords

Application Management, Network Management, Distributed System Management, Mobile Agent

1. Introduction

Most of the current network management technologies focus on collecting management information and providing a graphical-user interface to assist network managers in visualizing the collected management information and carrying out their management tasks (passive management). Furthermore, the type of information collected is not appropriate to achieve end-to-end proactive management functions. There has been little work done to make network

management systems proactive and intelligent. By making network management systems proactive, all management functions will be improved and the network can respond in a timely manner to any changes in application requirements and available resources. The concept of management system and programmable application management schemes has not been investigated thoroughly and is not well understood. The main goal of our research is to develop a management framework to achieve end-end proactive application centric management.

Network management products have taken a long road to the current state. They started with the Simple Network Management Protocol (SNMP) version 1 and then version 2 [1,3], followed by the Management Information Base (MIB) version I and then version II, and Remote Monitoring (RMON) version I and version II. Recently, there has been an intensive effort to use web-based technologies (JMAPI and WBEM) to build network management tools [6,7]. The limitations and problems with network management are:

- Most of the commercial network management systems collect management information about packet throughput, delay, and packet errors at input and output of network interfaces. For end-to-end proactive application and network management, we need to collect management information relevant to applications and computing resources such as the current loads on computers, the types of processes accessing the file systems, types of users and their access pattern profiles, security information, and so on. Furthermore, we need to have the ability to program the type of information to be collected and for what period of time. This approach enables us to dynamically collect any required management information and for the required period of time rather being running all the time and consuming unnecessary computing and storage resources.
- The amount of information collected for large networks (enterprise networks) is huge. It is very difficult for network managers to efficiently utilize the overwhelming amount of management information to improve network performance, utilization and applications. This problem becomes extremely complex when the size of the network increases to hundreds or thousands of nodes spanning several organization domains or countries. We need to develop techniques that can process raw management information and produce concise management information filters. These management filters can then be used to achieve efficient and robust analysis of large-scale networks and their applications.
- The literature is rich with algorithms and techniques to dynamically route packets, automatic reconfiguration, adaptive scheduling of resources, dynamic fault tolerance, etc. However, very little is done to apply these algorithms/techniques in order to achieve proactive real-time management of networks and their applications.
- Most of the management functions such as configuration, resource allocations and scheduling are done manually by network managers. This makes the management process slow, not scalable, and not cost-effective. Furthermore, this manual management scheme can not meet the stringent real-time requirements of some critical applications.

We do need to develop novel management techniques that eliminate these problems and provide scalable management capabilities to efficiently, intelligently,

and cost-effectively manage any network application running on any network of any size and at any time. In this paper, we present a framework for a network management system that provides management services to bridge the gap between application development and network management as well as build management ready applications. Our approach for the implementation of the network management framework is hierarchical and consists of three layers: Network and Protocol Management (NPM), Management Computing System (MCS), and Application Centric Management (ACM). The NPM is responsible for the collection of management information not only about the network devices, but also information related to computer processes, file systems, user access information and patterns, and protocols. The NPM will also perform tasks to manage the network devices, protocol functions, computer processes and file systems. The MCS provides the core management functions to manage system-wide resources from a system perspective rather than component perspective as is done in NPM. The ACM provides the capability to program MCS functions to control and proactively manage a given network application during all the life cycles of any network application. We present also the architecture of a Proactive Application Management System (PAMS) and evaluate the performance of some of the management services offered by the PAMS prototype.

The organization of the paper is as follows. In Section 2, we overview the current research of intelligent mobile agents. In Section 3, we describe our network management framework and the main software modules to implement this framework. In Section 4, we present the architecture of a proactive application management system implemented based on our management framework. In Section 6, we evaluate the performance of some of the management functions offered by PAMS prototype. In Section 6, we present a summary and concluding remarks.

2. Related Work On Agent-based Management

The current management systems are based on a platform-centered paradigm that separates applications from the required data and service. This centralized paradigm has severe limitations when applied to manage in real-time the emerging large scale, complex multi-domain networks. There is a growing need for new technologies to automate and distribute management functions that are required to achieve scalable and robust management. Mobile agents are programs that can be dispatched and executed remotely under local or remote control. There are two general approaches identified for agent-based service architecture: remote execution (or smart network) and migration (or smart message).

- **Remote Execution Approach**: In the remote execution approach, agents are static entities in the network and can perform tasks autonomously and asynchronously. These agents can also communicate with other agents and can be dynamically configured. The issue raised by this approach is the dynamic downloading and/or exchange of control scripts that means intelligence resides mostly at the network devices. The control scripts can be simple or complex and represent ``lightweight'' mobile agent(s).
- **Migration Approach**: In this approach, agents are mobile entities that travel between computers/systems to perform tasks. In this approach, intelligence is partitioned in a balanced way among the main management system, the agents and

the local environment. The migration agent can serve as an asynchronous message carrier for its owner, or as a broker that requests and sets the requirements of services.

With these agent-based approaches, services can be provided instantly, customized, and distributed [14,16]. In a scheme based on [17], generalized scripted and delegated agents for management has been proposed. The mobile agents are sent to remote sites where they are incorporated into the local network management program and are sued for intelligent tasks like MIB filtering. These are mostly remote execution agents and focused on monitoring the destinations. They are not used for delivering control information to configure the network topology.

Some agent implementations have been proposed and developed to manage various domains such as ATM connection management [18], and managing mobile connection based on client-server model [19]. The use of mobile agents to achieve effective and efficient network management is becoming increasingly important.

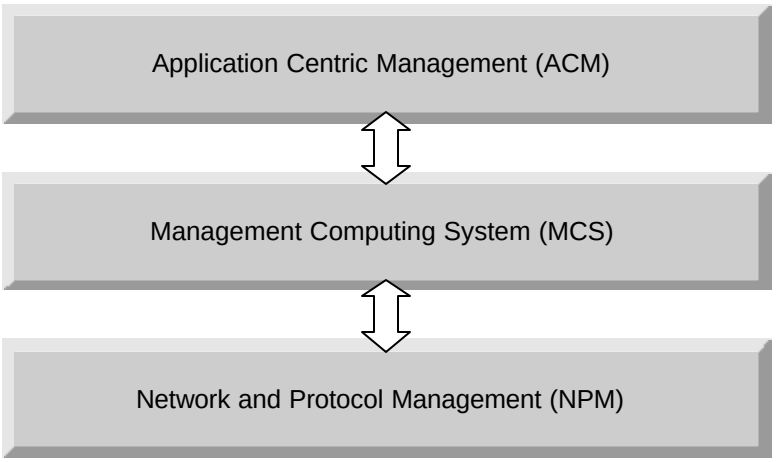

Figure 1 A framework for end-to-end proactive network management system.

3. Framework For Application Centric Management

The management framework we are developing can be viewed in terms of three systems: Network and Protocol Management (NPM), Management Computing System (MCS), and Application-Centric Management (ACM). The NPM is responsible to collect management information not only about the network devices, but also information related to computer processes, file systems, user access information and patterns, and protocols. The NPM will also perform tasks to manage the network devices, protocol functions, computer processes and file systems. The MCS provides the core management functions to manage the whole system resources from system perspective rather than component level perspective. In order to achieve that, the management information collected at the lower level (NPM) will be analyzed and abstracted into suitable data structures or format to

perform efficient system level management functions. The MCS design concept is analogous to the operating system in computing systems. The operating system manages the computing system resources (memory, I/O, CPU, and processes). Similarly, the MCS acts as an automatic system manager that provides management functions to achieve application centric management tasks.

The ACM provides two main functions: Assist in the development of application management routines, and provide intelligent proactive management for a wide range of network applications. The number and type of network applications become increasingly large and their computing, storage, and network requirements differ widely. In addition to the difficulty that can be contributed to the complexity, heterogeneity, and size of the emerging network applications, the development of such applications do not take into consideration the management issues and requirements. Currently, the management of such applications follows force-fitting approach that attempts to rely on the commercial network management services that are based on SNMP or CMIP standards.

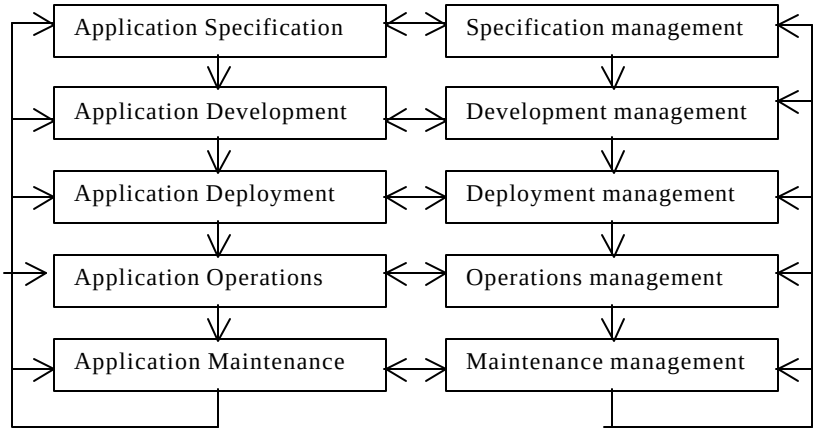

Figure 2. Software Development cycle with Management Activity

Our approach is to develop system management functions (provided by the MCS) that can be programmed by applications to meet their requirements during all the life cycles associated with any application (e.g., specification, development, deployment, operations, and maintenance). Figure 2 shows how we can integrate the software development of an application with the management activities of that application.

4. Architecture of a Proactive Application Management System (PAMS)

In this section, we first overview the architecture of a Proactive Application Management System (PAMS) being implemented at the HPDC Laboratory at the University of Arizona. Then, we benchmark the use of mobile agent technologies to implement important PAMS modules. The architecture of PAMS is shown in Figure 3. The main key components of PAMS can be described in term of three services: ACM Service, MCS Service, and NPM Service.

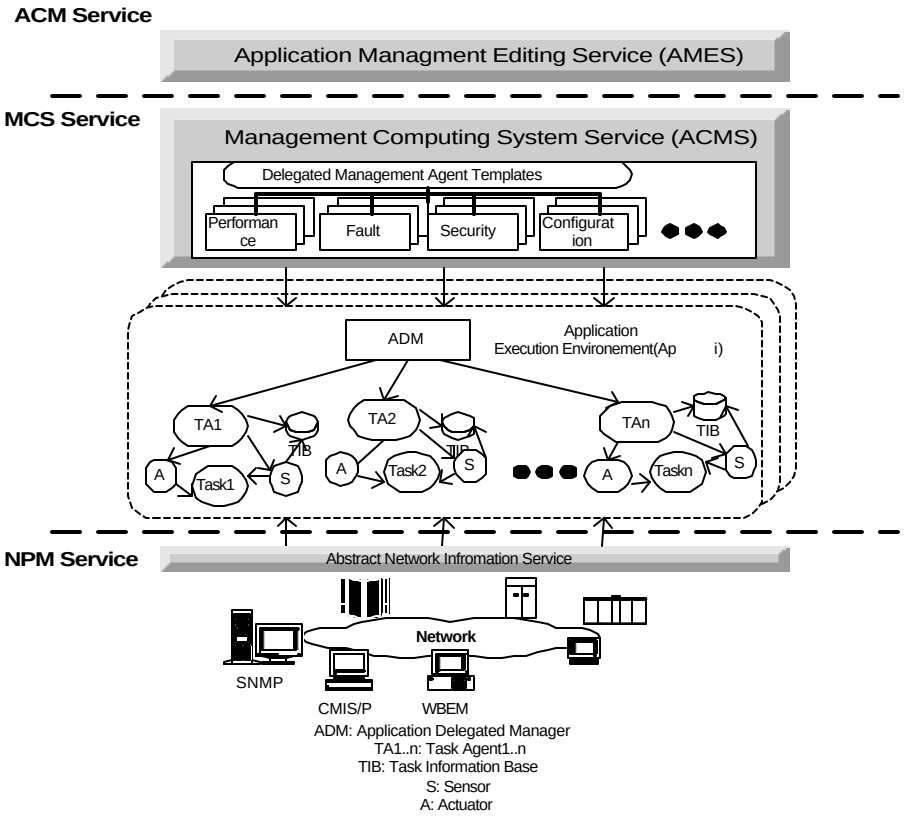

Figure 3. The architecture of PAMS

The ACM Service provides the user with the tools required to describe and characterize the management requirements of any network-centric system or NCS application. The MCS provides the management services to automatically configure the application or system resources, monitor and control the execution of an NCS application. The NPM service provides the appropriate interface to existing network management systems and utilize their services in order to proactively manage and control the operations of the NCS or its applications.

When a user develops an application using PAMS user interface subsystem, the application requirements are described and characterized by an application graph as shown in Figure 3. The MCS sever checks the Management Agent (MA) templates and generates the appropriate Application Delegated Manager (ADM) appropriate for application performance, fault, security and configuration. Once the ADMs required to execute and manage the application are identified, the next step is to download the Task Agents (TAs) and the appropriate execution codes into the selected computing resources. On each machine selected, a

Task Agent (TA) is activated in order to start the execution of the monitored task on that machine.

4.1 The MCS Services

To support PAMS proactive management services, the MCS services support the following four categories of ADM services:

- Configuration: The Configuration services locate and discover resources that are accessible through the PAMS networks and maintain the required management information and network topology.
- Performance: The Performance services collects management information for any application and/or resource. The Performance service can monitor the following attributes:
  - Responsiveness: The responsiveness of an application or a managed resource is the time between an operation invocation and the completion of the operation. For example, responsiveness of a managed object can be the response time of a request, completion time of a workload and process initialization time.
  - Productivity: The productivity of a managed object represents the amount of work completed per unit of time. Productivity examples include the number of requests processed per second in a server and the amount of data transferred between tasks per second.
  - Utilization: The Utilization indicates the level of system resource consumption. Utilization examples include the utilization of CPUs, memory system, Disk, I/O, and network.
- Fault: The Fault service monitors the conditions of failure in which an application operation cannot be completed due to hardware and/or software failures and provides various strategies to recovery the problems.
- Security: The Security service is responsible of secure execution of an application execution and guarantees the access to appropriate resource for the application.

The MCS services can be implemented in a hierarchical manner when the ADM task involves the management or monitoring a large number of resources spanning several domains. For example, we could have several ADMs to manage the performance of each managed object. The TAs interact with managed objects through well defined interfaces that we refer to as Sensors and Actuators. The sensors provide the appropriate interface to determine the status of performance and/or fault of any managed objects. The Actuators provide the interface to enable the TA to terminate, suspend, resume or migrate the execution of a managed task. Figure 4 shows the functionality to be supported by PAMS sensors and actuators.

5. Evaluation of PAMS Management Services

In this section, we evaluate the use of PAMS ADM and TAs to perform local computations, error control and fault tolerance functions at the managed objects location.

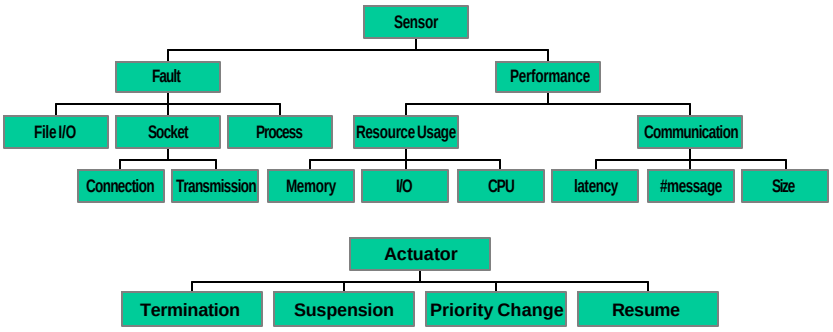

Figure 4. The functionality of PAMS sensors and actuators.

5.1 Using the ADM to Perform Traffic Analysis

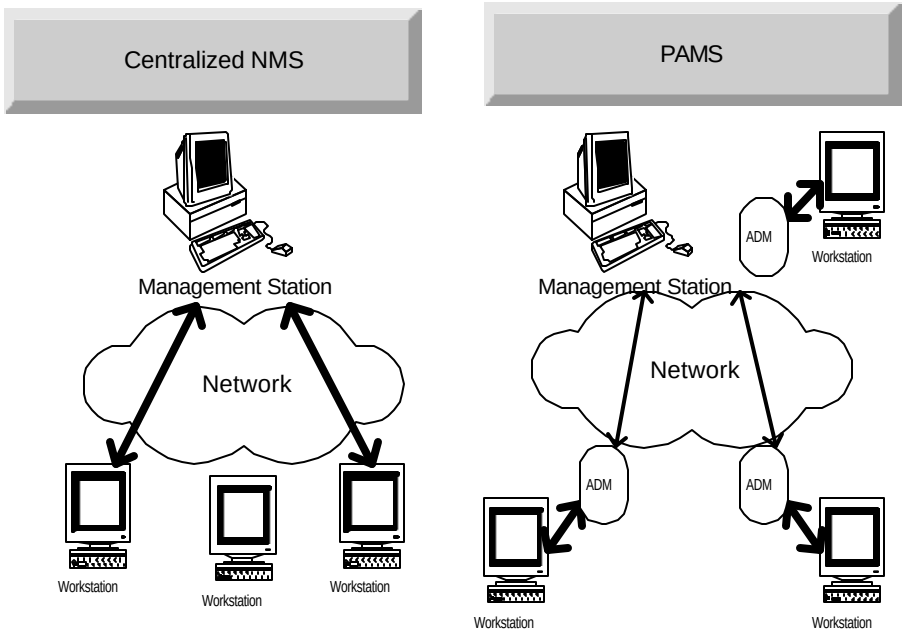

Figure 5: Experimental Environment

In this experiment, it is required to evaluate and monitor continuously the amount of traffic between consecutive intervals, and displays the amount of traffic

changes between them. Using SNMP application, the management station needs to obtain traffic for each interval nad must be repeated for all the managed objects, i.e.

Managed Object:
for each managed object **do**
obtain fraffic_value(t) from all interfaces
Send to the management station
End for
Management Station:
Receive all traffic value(t)
get the difference between traffic_value(t) - traffic_value(t-1)
Display the results

For *n* resource, the number of SNMP messages required is *n* * *Pmsg** *Tmsg*, where *Pmsg* is number of SNMP messages, and *Tmsg* is the transfer time of a message through the network.

To achieve the same type of analysis using PAMS TAs, the following is required.

Managed Object:
Load Task Agent on each managed object
Each agent computes the following:
- traffic_difference(t) = traffic_value(t) – traffic_value(t-1)
- Send traffic_difference(t) to the management station
Management Station:
Receive the results
Display the results

The number of messages required to pass the results is *n* **Tmsg*. In addition, the processing time on the server is much faster since the computation is distributed at each managed resource. Figure 6 shows the traffic summary between consecutive periods that are calculated by the local agent, and displayed by the central management station.

5.2 Using the ADM for Error Detection

In this experiment, it is required to monitor the amount of error rate at each interface. We compare the bandwidth required to achieve this managed task using our approach and compare against the SNMP approach. To monitor the error using SNMP agents, the following routine is needed:

For each managed object i **do**
- Obtain the error rate at each interface through the network
- **if** Error_rate (i) > Error_threshold **then**
send message to the central management station to disable the interface

end For

Using PAMS ADM, the following routine is needed:

Each Task Agent will execute the following routine:
- monitor the error rate using sensors
- **if** the error rate > Error threshold **then**
report error to the management station

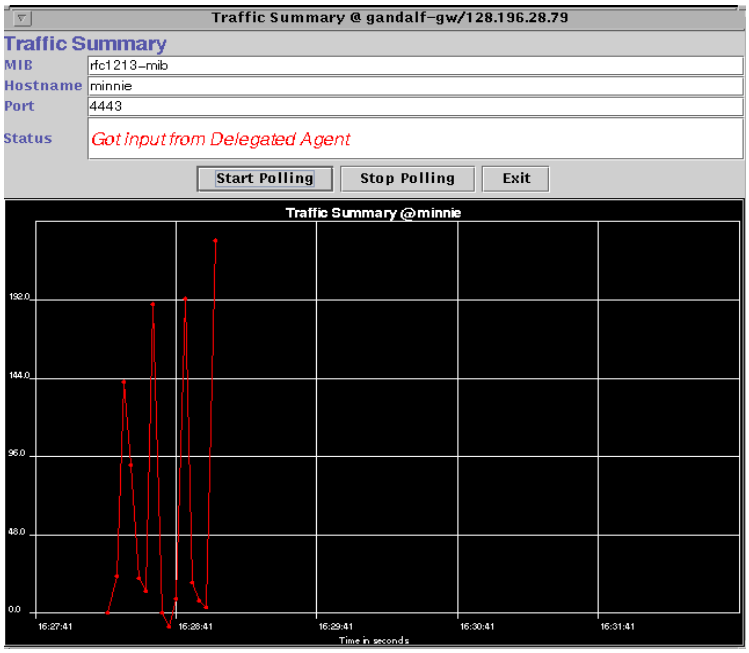

Figure 6 The Result of Traffic Summary

Figure 7 The Result of Error Detection

The number of messages required in SNMP approach is *n* * *Pmsg* * *Tmsg* since checking for errors is performed in the centralized management station. In PAMS approach, the number of messages is *Tmsg* when there is an one error. However, there is no traffic overhead when there is no error detected in the system. In addition, the processing time is much less in PAMS approach because the computations are performed locally at each node rather than being implemented on the management station. Figure 7 shows the results produced by the agent when to detect an error on an interface is above the threshold.

5.3 Automation of Application Fault Management

In another experiment, we used one ADM to achieve fault tolerance execution of a managed task. We focus on the automation of application fault management. Currently, we are building up the architecture of the management system, which allows users to design management tasks to support a specific application requirement (e.g., fault-tolerance capability). The fault tolerance algorithm in the example is to migrate the faulty task to another fault free machine.. The task has intelligence to cope with any unexpected events and provide transparent management to the application execution. Like Application MIBs [21][22], we define simple TIB (Task Information Base) to hold user requirement, system requirement, execution and management. We monitor the execution of a given application and when the application is crashed/killed, the manager provides proactive action by assigning a new machine to maintain the fault tolerance requirement. The overhead incurred in error detection and recovery using PAMS ADM service and without PAMS (by manually detecting and recovering from the error) is shown in Figure 6. The average recovery time after detecting application fault is 50 milisecond. As it is impossible to implement with existing NMS, we claim that it is a significant experiment to manage an application fault.

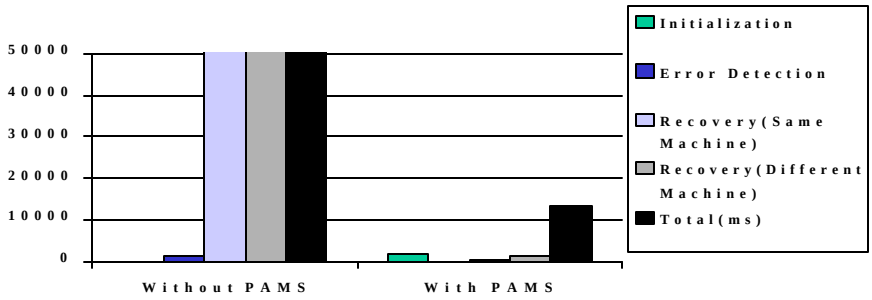

Figure 6. Evaluation of PAMS fault tolerance service.

6. Summary And Conclusion Remark

In this paper, we presented a general approach to develop application centric management. We also described an architecture to achieve proactive application management. Our approach is scalable and utilizes delegated agent approach to implement distributed management services. We have also compared our approach with SNMP approach when applied to analyze traffic summary, error rate management and automation of application fault management. Our preliminary results show that PAMS approach can lead to better performance when applied to manage in real-time large-scale network-centric systems and their applications. We are currently extending the capabilities of PAMS to manage security. We are also implementing the ACM tools to allow users to describe the management routines required to manage any management function.

7. REFERENCE

1. J. Case, M. Fedor, M. Schoffstall, and C. Davin, "Simple Network Management Protocol (SNMP)," RFC 1157, May 1990
2. J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Introduction to version 2 of the Network Management Framework," RFC 1441, April 1993
3. K. McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets," RFC 1156, May 1990
4. K. McCloghrie and M. Rose, "Structure and Identification of Management Information for TCP/IP-based Internets," RFC 1155, May 1990
5. K. McCloghrie and M. Rose, "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II," RFC 1213, March 1991
6. Sun Microsystems, Inc. "Java Management API Home Page," http://java.sun.com/products/JavaManagement/index.html, November, 1996
7. Microsoft corporation, "Web-Based Enterprise Management Initiative Home Page," http://wbem.freerange.com, 1997
8. Microsoft corporation, "Microsoft Web-Based Enterprise Management Professional Developers Kit." 1997
9. S. Waldbusser, "Remote Network Monitoring Management Information Base," RFC 1757, Feb. 1995
10. A. Franceschi, L. Kormann, C. Westphall, "Performance Evaluation for Proactive Network Management," IEEE 1996
11. M. Jander, "Welcome to the Revolution," Data Communication, Nov. 1996
12. T. Yang, "Performance Analysis on Distributed Network Management System," IEEE 1996
13. R. Voruganti, "A Global Network Management Framework for the 90s," IEEE 1994
14. T. Magedanz, T. Eckardt,"Mobile Software Agents" A new Paradigm for Telecommunication Management", NOMS96, Kyoto, Japan, April, 1996
15. I. Ahmad, A. Ghafoor, K. Mehrotra, C.K. Mohan, and S. Ranka, "Applying Neural Networks to Performance Evaluation of Dynamic Load Balancing Algorithms," IEEE Concurrency: Practice and Experience, Vol. 6, No. 5, pp. 393-409, Aug. 1994

16. T. Magedanz, “Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?,” IEEE INFOCOM, San Francisco, California, USA, March, 1996.
17. G. Goldzmith, and Y. Yemini, “Decentralizing Control and Intelligence in Network Management,” In the Proc. Of 4th International Symposium on Integrated Network management, Santa barbara, 1995
18. D. Hall, S. Rooney, ”Controlling the Tempest: Adaptive management in Advanced ATM Control Architecture,” IEEE Journal on Selected Areas in Communications, vol 16, No. 3, April, 1998
19. A. Sahai, C. Mortin, S. Billiart, “Intelligent Agents for a Mobile Network Manager(MNM),” 1997 Intelligent Networks and Intelligence in Networks, 1997
20. S. Hariri,D. Kim, Y. Kim, et.al "Adaptive Distributed Virtual Computing Environment(ADViCE)" submitted to The Transaction of Software Engineering Journal, 1998
21. C. Krupczak and J. Saperia, Definition of System-Level Managed Objects for Applications, RFC2287, Feb 1998.
22. C. Kalbfleisch, C. Krupczak, R. Presuhn, and J. Saperia, Application Management MIB, Internet-draft, Nov. 98.