A Framework for Network Management using Mobile Agents*

Manoj Kumar Kona and Cheng-Zhong Xu
Department of Electrical and Computer Engineering
Wayne State University, Detroit, MI 48202
{manoj, czxu}@ernie.eng.wayne.edu

Abstract

Conventional network management is based on SNMP and often run in a centralized manner. Although the centralized management approach gives network administrators a flexibility of managing the whole network from a single place, it is prone to information bottleneck and excessive processing load at the manager and heavy usage of network bandwidth. For scalability, this paper presents a mobile agent based network management framework (MAN). Mobile agents for network management tend to monitor and control networked devices on site and consequently save the manager capacity and network bandwidth. The MAN architecture is a hybrid model. It provides an additional network management interface to administrators atop SNMP layer so that the administrator can have the flexibility of using a most appropriate management approach according to the network characteristic and nature of management activities. The MAN architecture has been prototyped based on an AdventNet SNMP package and an IBM Aglet mobile agent system. The MAN system has been experimented with in a network interface performance monitoring application and a fault diagnosis application. Experimental results have demonstrated the advantages and disadvantages of each method and the flexibility of the MAN framework.

1 Introduction

Network management essentially involves monitoring and controlling the devices connected in a network by collecting and analyzing data from the devices [20]. Due to the increasing use of heterogeneous computing environments, network management becomes more and more difficult. Users' increasing expectations for reliability and quality of service make it even harder.

Conventional network management is based on SNMP (Simple Network Management Protocol) for large networks [19]. It gives network administrators the flexibility of managing the whole network from a single place. SNMP is often used in centralized network management environments. Its drawbacks include information bottleneck at the manager, lack of scalability, excessive processing load at manager, heavy usage of network bandwidth by network management actions, management intelligence too centralized.

An alternative is distributed network management, in which the centralized management strategy is replaced by interoperable management systems. Distributed management solves the problems with centralized management to some extent. However, it still has some drawbacks like limited scalability and complex coordination mechanisms between management stations.

The latest trend is to deploy mobile agents to manage today's large heterogeneous networks. Mobile agents are special software objects that are autonomous and have the ability to migrate from one node to another node, carrying logic and data, performing actions on behalf of the user. Mobile agent based network management is to equip agents with network management capabilities and allow them to issue requests to managed devices (or nodes) after migrating to these nodes.

Mobile agent based network management has many advantages. We present them with regard to functional areas of network management: performance, configuration, accounting, and fault management. Performance management involves gathering statistics about network traffic and schemes to condense and present data. Measuring performance of networks using centralized SNMP based management is very difficult due to reasons like network delays and information bottleneck at the central management station. The solution to performance management using mobile agents is superior to conventional management techniques. Mobile agents give the flexibility of analyzing the managed node locally.

Instead of querying the managed node for every fixed interval and analyzing the performance from management station, MA can be dispatched to analyze the node locally. Advantages with mobile agent based performance monitoring include:

- Information collected using this approach is more accurate since no delays are involved.
- Local manipulation of data on the node.
- Filtering operations could be performed on managed node. This condenses and organizes data before it brings the analysis back to the network management station.

* This research was supported in part by NSF grants CCR-9988266 and EIA-9729828.

Configuration management is concerned with initializing a network, service provisioning, and adding and updating relationships among components and status of components during operation. Mobile agents can help smoothen configuration management which otherwise is a complex process. Configuring a device, which is connected in a network, requires setting a number of parameters and installing software components. Currently the manager has to perform all the tasks manually. A mobile agent could be equipped with the necessary parameters and software and dispatched to the agent that needs attention regarding configuration. This approach is more efficient in the sense that it helps in disconnected operation between management station and managed node.

Accounting management is the process of gathering statistics about the resources of on the network, establishing metrics, checking quotas and determining costs. The same strategies, as described above, could be used for achieving proper accounting management.

Fault monitoring involves identifying faults in devices connected to a network. Detection of faults can be achieved through the use of mobile agents. A mobile agent performing analysis of devices connected in a network could be programmed to have the ability to report to the management station about all nodes whose utilization increases a certain threshold. In some cases we need to rely on low-level protocols for detection and correction of fault. Since our flexible architecture doesn't loose the advantage of conventional SNMP management techniques, low-level fault detecting schemes based on SNMP could be used in the detection and correction of some faults.

To take advantage of mobile agents for network management, we propose a flexible architecture in which Mobile Agent based Network Management (MAN) forms a layer over conventional SNMP based management. This ensures that the advantages of SNMP are not lost and also serves the purpose of managing legacy SNMP based systems. We believe that mobile agents and SNMP based management should coexist in building an efficient management system.

While mobile agent technology has long been pursued, its applications in network management are still rudimentary. The Carleton effort [17] is commendable wherein mobile code was used for managing the network. Network manager is a suite of simple tools that interact with agents located on nodes connected in a network, as well as other tools used for managing networks. Each of the tools can be used for actual management of networks. Less stress was laid on embedding existing SNMP into a mobile code framework. This ensures incompatibility with SNMP based systems. Bieszczad et al. [6] described theoretical views on application of mobile agents for network management. Since much of the research is in infancy, so the authors illustrated application ideas in several areas of network management through their own experience. They explained the application of mobile agents for network management but lacks concrete implementation. Gavalas et al. [10] presented the application of mobile agents in bulk transfers of network monitoring data, data aggregation and acquiring atomic SNMP table views. They analyzed the usage of mobile agents in network management with regard to the bandwidth utilization. This work addressed the issue of mobile agents for network monitoring, but did not consider SNMP related compatibility issues. Pinheiro et al, [21] described a conceptual model which collects management related data across a changing set of networked components and periodically computes aggregated statistics using mobile agents. More concentrated towards aggregation of network monitoring data and exploring mechanisms for agent adaptation.

The paper is organized in the following way. We first introduce concepts involved in SNMP based network management and then present the MAN framework and its components in Section 2. In Section 3, we describe network management patterns using mobile agents and demonstrate the effectiveness of their usage in determination of a "Health Functions". Section 4 presents experimental results about the MAN. We conclude the paper in Section 5 with remarks on future work.

2 MAN: A New Network Management Framework

SNMP is the dominant protocol for network management. SNMP supports the operations: *Get-Request; Get Next-Request Get-Response, Set-Request, Trap*. In SNMP a management application uses the manager protocol to communicate with the managed system, which uses the agent protocol to communicate with the MIB and the manager protocol. This model is often used to manage whole network from a central management station. Processing of managed data is done at the management station. Network management stations interact with SNMP agents in managed nodes. Each SNMP agent is essentially a daemon process that responds to requests from management stations. SNMP agents are organized in different ways in different platforms.

Our MAN framework is a hybrid model, which has features of mobile agents as well as SNMP. MAN gives the manager the flexibility of using SNMP model or mobile agent based management depending on the management activity that is involved. This architecture has many advantages over the existing architectures.

2.1 Architecture and Components

Figure 1 shows the MAN architecture for network management using mobile agents. The manager is given

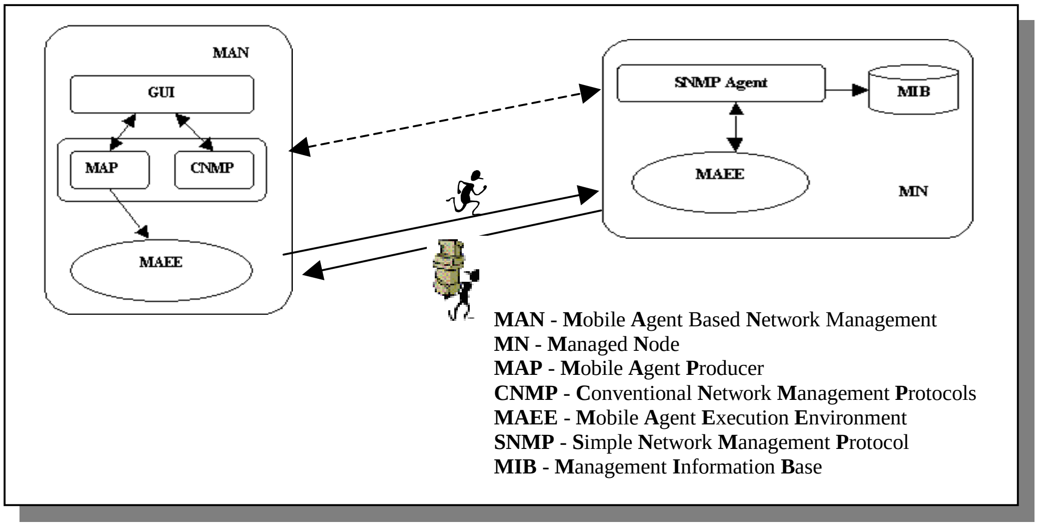

Figure 1. Architecture of our approach

the flexibility of deciding whether to use SNMP or mobile agents.

In this approach the MAN station assumes responsibilities of a client. All managed nodes are servers, which have mobile agent execution environment and respond to SNMP queries from mobile agents when they visit the servers and manipulate data locally. When the client in the MAN needs access to data on a network-connected device, it does not talk directly to the server over the network. Instead, the client actually dispatches a mobile agent to the server's machine. Once on the server's machine, the MA makes its requests to the server directly. When the entire transaction is complete, the mobile agent returns to the management station with the results.

The MAN provides Java-compliant interfaces to network management services. The MAN itself was developed in Java as well because of Java's write-once-run-everywhere commitment and its dynamic class loading and object serialization features [13]. The MAN framework consists of the following major components:

Management Application

The management application has a Graphical User Interface (GUI), which coordinates with the agent applications underneath it. It interacts with Mobile Agent Producer (MAP) in configuring a MA with details such as the parameters to be evaluated at the managed node's site and *health functions* [5].

Mobile Agent Execution Environment (MAEE)

MAEE is an execution environment for the execution of MA's. MAEE could be characterized as home for mobile agents from where they could execute their duties. The agent comes to the managed node from the management station, executes its management task and goes back to the management station. We are using *Aglet Server* as our MAEE. Aglet server is the mobile agent execution environment in machines that host mobile agents. In MAN, MAEE acts as an interface between MA's and SNMP agent at the managed node. Figure 5 depicts the internal view of an Aglet Server [8].

Mobile Agent Producer (MAP)

MAP could be characterized as a tool for generating customized MA's that are equipped according to the requirements of network manager. By using MAP the functional characteristics of MA's, which roam in the network to collect information from managed nodes, can be changed dynamically (i.e. at runtime). Dynamic creation and configuration of MA's is achieved using MAP.

Mobile Agents (MA)

A typical mobile agent is Autonomous, Mobile, Persistent, Communicative/ collaborative, Active /proactive. The ability to travel allows mobile agents to move to the network element which is to be managed. In other words, mobility of MA's could be exploited to transfer the MA to managed node and interact locally with the SNMP agent on the managed node.

In MAN framework mobile agents are provided with:

- The list of nodes to be managed.
- SNMP statistics of interest.
- Health functions defined by the user.

We uses Aglets as the agent development environment. *Aglets Software Developer Kit (ASDK)* provides a modular structure, easy-to-use API for programming of mobile agents and excellent documentation.

Conventional Network Management Protocol (CNMP)

As mentioned, conventional SNMP based management is embedded into our architecture to ensure compatibility with SNMP based systems. This ensures that advantages of lower level protocols are not lost. To interact with the SNMP agent we have used AdventNet SNMP[1]. AdventNet SNMP provides a set of Java tools for creating cross platform Java and Web-based SNMP network management applications. This package provides a set of classes, which could be used to facilitate communication between a managed device (a device with an SNMP agent, e.g. a router), and an SNMP manager or management application.

Using the mobile agent paradigm can bring some interesting advantages when compared to traditional client/server solutions [10]. It can reduce network traffic; it can provide more scalability; it allows the use of disconnected computing; and it provides more flexibility in the development and maintenance of the applications. Intelligent information retrieval, network and mobility management, and network services are three most common application targets for a mobile agent system.

2.2 MAN over TCP/IP

The architecture of MAN demonstrates mobile agents as software entities that roam in the network over TCP/IP. The operation of MAN over TCP/IP suggests a healthy advantage of Mobile Agent based management.

Using conventional management techniques, retrieving large amounts of MIB data involves a high number of Protocol Data Unit (PDU) exchanges over network. To improve the above solution RFC1187 describes an algorithm that speeds up the retrieval of an entire table in a MIB by using multiple threads in parallel, where each thread retrieves a portion of table. But this algorithm achieves reduced latency at the cost of the following disadvantages:

- To make above algorithm work a manager which supports multiple threads and which has knowledge of distribution of instance identifiers in MIB is required.
- Situation could get worse in case packets get dropped with in network.
- Multiple threads cause bursty SNMP traffic on the managed node.

Another operation, Get-bulk operation could be used to perform bulk retrieval of MIB data, but still Get-bulk operation has to fit into a single UDP packet, since SNMP works over UDP. For grouping large amounts of MIB data which might be in the order of hundreds of kilobytes, results in large overall delay because UDP packets can handle packets of size less than or equal to 64 kilobytes.

The state carrying and state restoring capability of mobile agents could be exploited to carry management data. Mobile agents can be configured to procure data from different tables of MIB which is selective retrieval of MIB parameters, where as Get-next and Get-bulk operations are supported over well-related subsets of data in MIB. For example Get-next can be used only to get the next row of the current interaction with MIB table. If management involves large amounts of data and multiple machines MAN reduces end-to-end latency.

Philippe and Sprinkles mentioned the advantage of adding SNMP over TCP/IP in “Bulk transfers of MIB data” [18]. They mentioned that the effect of moving from UDP to TCP removes the limitation of maximum SNMP message size of 64 kilobytes. Mobile agents, which work over TCP/IP, could be used for network management. “MAN” suggests an architecture in which mobile agent based management forms as a layer over conventional SNMP based management. This also ensures that advantages of SNMP based management are not lost. So the Network Manager can be given the flexibility of selecting UDP or TCP based on the amount of management data that is involved. When a manager needs small amount data from a small set of managed nodes, UDP is a better choice. When a large amount of MIB data is to be retrieved from multiple managed nodes mobile agent based management is desirable.

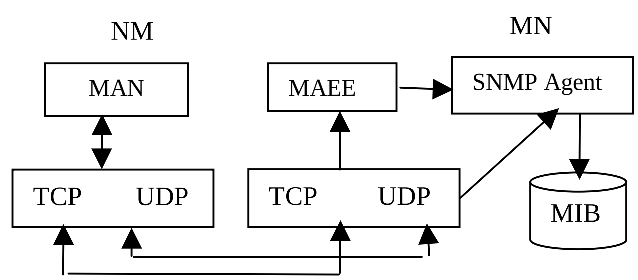

Figure 2. MAN over TCP and UDP

3 Travelling patterns for mobile agents managing the network

The possible MA travelling patterns which we designed while managing a set of managed nodes are itinerary (roaming) model and broadcast model. In the itinerary model the MA generated by MAP has the list of all the managed nodes to be visited.

3.1 Broadcast Model

In this scheme a MA is dispatched to each managed device (see Figure 3). All the dispatched MA's stay at their respective node and analyze it for amount of time specified by the MAP. MA's poll the managed nodes after each polling interval (specified by MAP). Each MA agent stays there for an amount of time equal to the total number of polling intervals. It executes its task by interacting with the SNMP agent for each polling interval, performing necessary calculations on obtained management statistics, analyzing them by using some functions equipped into the MA and get back to the manager.

$IfOutOctets_x$ is the bytes sent by the interface at time x, (y-x) is the polling interval.

By using broadcast scheme we could dispatch MA's to each of the managed nodes and MA's could calculate the utilization for every polling interval over an extended period of time. Here the MA's manipulate the data locally at the managed node. After the time period for which the analysis is required, which is equal to the sum of all the polling intervals, the MA returns to the management station and the reduced data set could be displayed in a graph. Here MAs are used to represent a reduced data set

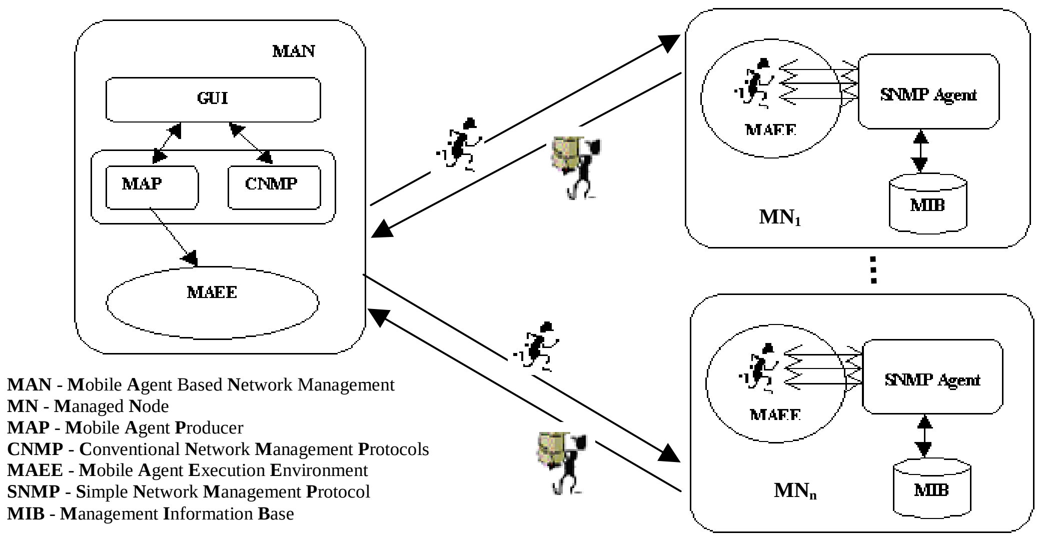

Figure 3. Broadcast Mobile Agent approach

An example illustrating the use of the above approach is described as follows. This scheme could be used to monitor the performance of a set of managed nodes over a particular interval of time. There are cases that involve a set of management parameters in calculating a cumulative factor. Such cumulative are called Health Functions, which indicates system state or efficiency of the node and could be viewed as a way to compress management data and evaluate the performance of any element.

A performance management application can use ifInOctets and ifOutOctets of the interfaces group in MIB to compute the percentage utilization of an interface over an interval of time. To perform this computation, two different polling intervals are required: one to find total bytes per second at time x and another to find total bytes per second at time y. The following equation computes utilization, U(t) for the polling interval (x-y) seconds [5]:

$$U(t) = \frac{((ifInOctets_y - ifInOctets_{x)} + (ifOutOctets_y - ifOutOctets_y))*8}{(y-x)*ifSpeed},$$

where, *ifspeed* is the bandwidth of the interface, *$inOctets_x$* is the bytes received by the interface at time x,

polling intervals, the MA returns to the management station and the reduced data set could be displayed in a graph. Here MA's are used to represent a reduced data set and removes information bottleneck and processing loads at the management station.

In contrast, if conventional client/server architecture is used in calculating utilization at interfaces of a set of nodes, the following operations are to be performed on each managed node from the management station. For each of polling interval,

1. Send get-request for *$ifInOctets_x$* and *$IfOutOctets_x$* and receive response from the managed node.
2. After polling interval send get-request for *$ifInOctets_y$* and *$ifOutOctets_y$* and receive response from the managed node.
3. Calculate the total bytes per second and utilization at the managed node.

With multiple machines to be monitored, the amount of processing overhead and information bottleneck at the management station would be high.

3.2 Itinerary model

Itinerary model is described as roaming management model. In this scheme a mobile agent visits the set of nodes to be managed sequentially.The mobile agent is configured with the list of nodes to be visited during its itinerary and also the SNMP statistics to be analyzed. configuration of agents is done while MAP dynamically creates the agent at network management station. A mobile agent sequentially visits all nodes to be managed sequentially.At each managed node it obtains required statistics, performs necessary calculation's in analyzing statistics (reduces the amount of management data that it would carry) before it visits the next managed node.

Health function is used to calculate percentages of input and output errors on an interface. It is a cumulative factor of 8 MIB variables [5]:

Percent input errors = ((ifInErrors)/(total packetsreceived))*100,
Percent output errors = ((ifOutErrors)/(total packets sent))*100,

where
total packets received = (ifInUcastPkts + ifInBroadcasts + ifInMulticasts),
total packets sent = (ifOutUcastPkts + ifOutBroadcasts + ifOutMulticasts),

This health function is often used in fault monitoring. If the interface error rate is more than 1%, then there is a problem with the interface of the machine. If the error rate is less than 1% and network shows poor performance, then it could be deduced that there is a problem with the media.

If monitoring of percentage error rates for interfaces of multiple machines is required an itinerary based MA could be dispatched, which sequentially visits all the machines. At each managed node it interacts with the SNMP Agent, calculates percentage input and output errors locally on the managed node. After visiting all the nodes it returns back to the manager. If conventional SNMP based management was used for the calculation of above health function's it introduces processing burden at the management station and network load.

4 Experiments and Performance Analysis

Experiments were conducted on a set of networked SUN UltraSPARC workstations. Tests were conducted to study end-to-end latencies of SNMP and mobile agent based strategies. Figure 10 plots the end-to-end latencies for fetching various numbers of management parameters from a set of 5 managed nodes, using SNMP and broadcast model of mobile agents, respectively. Results procured through SNMP based polling shows gradual increases of the end-to-end latency. Broadcast model shows superior performance as the number of parameters to be fetched from managed node increases. The plots in Fig. 4 are in agreement with what we argued in Section 3.2. That is, when a manager needs small amount data from a small set of managed nodes, SNMP is a better choice and that when a large amount of MIB data is to be retrieved from multiple managed nodes mobile agent based management is preferred.

Figure 5 shows how well a MAN works when compared with multithreaded program. We depicted the results using both green threads as well as native threads. The advantage of native threads is that they take the advantage of underlying multiprocessor environment. So native threaded code involves lower latency than green-threads. Figure 6 shows the performance of itinerary based mobile agent model versus SNMP based approach.

The above graphs clearly justify our view that mobile agent based management involves lower latencies and is efficient. As the number of parameters to be monitored increase, MAN shows superior performance over multithreaded as well as SNMP based approaches.

From the figure, we can see that the itinerary model leads to higher end-to-end latencies. However, it is efficient in terms of bandwidth utilization. Itinerary model would be effective when health functions are to be evaluated on managed node for a large time interval. These results encourage the combined use of MAN and SNMP for network management.

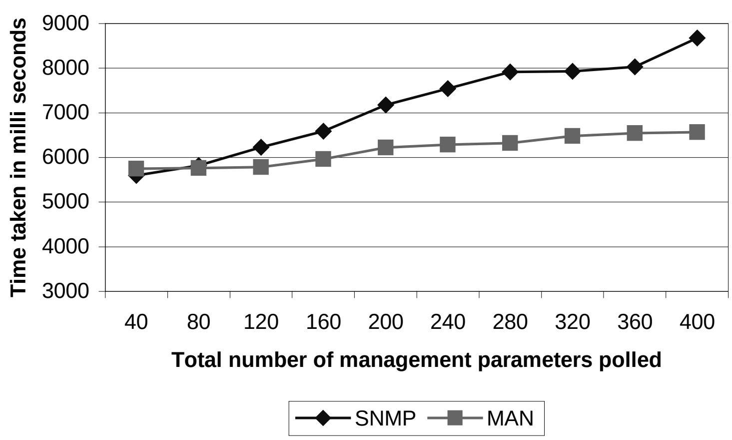

Fig. 4. Network Management Latency using SNMP versus Mobile Agent Broadcast Model

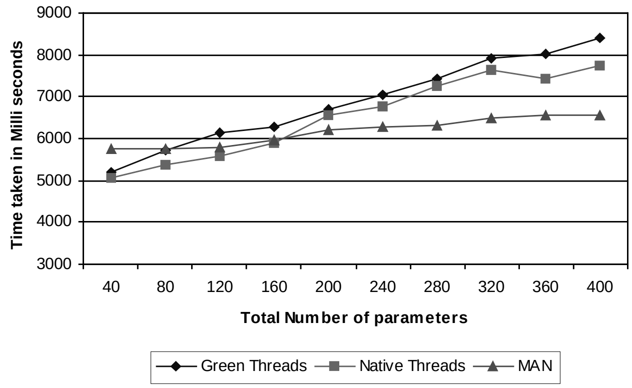

Fig. 5. MAN versus Multithreading

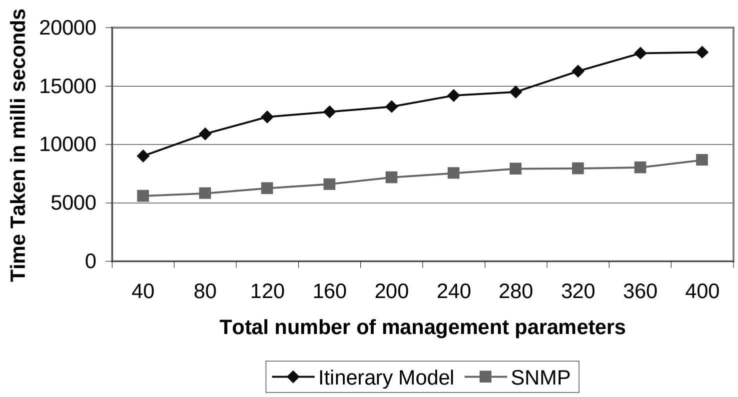

Fig. 6. Network Management Latency using SNMP versus MAN' s Itinerary Model

6 Conclusions and Future work

We have presented a framework for efficient management of heterogeneous networks. It is a hybrid model based on mobile agent and SNMP strategies. The framework gives management stations or network administrators flexibility of using any approach according to the characteristics of target networks. To exploit the potential of mobile agent technology for network management, we have proposed two agent-traveling patterns and discussed their usage, advantages and disadvantages in the evaluation of network health functions.

We plan to investigate into details related to the management of devices which cannot embed a mobile agent execution environment like hubs, switches etc. Evaluate the cost effectiveness of SNMP as well as mobile agent based approach. We also plan to extend the list of features in our framework.

References:

[1] The Advent Net Library. http://www.adventnet.com.

[2] Ajanta. A java-based mobile agent system. http://www.cs.umn.edu/Ajanta.

[3] Akhil Sahai, Christine Morin. Towards Distributed and Dynamic Network Management.

[4] Akhil Sahai, Christine Morin, Stephane Billiart. Intelligent Agents for a Mobile Network Manager (MNM).

[5] Allan Lienwand, K. Fang Conroy. Network Management: A Practical Perspective. Addison Wesley 1996.

[6] Andrzej Bieszczad,Bernard Pagurek and Tony White. Mobile Agents for Network Management. IEEE Communications survey-98'.

[7] Antonio Carzaniga, Gian Pietro Picco, Giovanni Vigna. Designing Distributed Applications with Mobile Code paradigms. Italy.

[8] Danny B.Lange / Mitsuru Oshima. Programming and Deploying Java Mobile Agents with Aglets. ADDISON-WESLEY 1998.

[9] D. Lange, M. Oshima. Seven good reasons for mobile agents. Communications of the ACM 42 (3) (1999) 88-89.

[10] D. Gavalas, D. Greenwood, M. Ghanbari and M.O. Mahogany. Advance network monitoring applications based on mobile intelligent agent technology. Computer Communications 2000.

[11] General Magic. Odyssey web page. http://www.genmagic.com/agents/odyssey.html

[12] Grasshopper – the agent development platform. http://www.grasshopper.de/index.html

[13] IBM Aglets software development kit. http://www.trl.ibm.co.jp/aglets/

[14] Management Information base for Network Management of TCP/IP based networks RFC1213. http://andrew2.andrew. cmu.edu/rfc/rfc1213.html

[15] Mario Baldi, Gian Pietro Picco. Evaluating the tradeoffs of Mobile Code Design paradigms in Network Management Applications. Dip. Automatica e Informatica, Italy.

[16] Network efficient mobile agent applications using concordia.http://www.meitca.com/HSL/Projects/Concordia/whatisit.htm.

[17] Perpetuum Mobile Procura Project at Carleton University. http://www.sce.carleton.ca/netmanage/perpetum.shtml

[18] Ron Sprenkels, Jean-Philippe Martin-Flatin. Bulk Transfers of MIB Data. http://www.simpletimes.org/pub/simple-imes/issues/7-1.html

[19] Sidnie Feit. SNMP : A guide to Network Management. McGraw-Hill, 1993.

[20] William Stallings. SNMP, SNMPv2 and RMON: Practical Network Management. Addison-Wesley 1999.

[21] Robert Pinheiro, Alex Poylisher,Hamish Caldwell. Mobile Agents for Aggregation of Network Management Data. Telecordia Technologies, University Of Warwick, BellSouth Telecommunications.