Reducing Response Time in Network Management Tasks by Using Mobile Agent Strategies

Marcelo Gonçalves Rubinstein and Otto Carlos Muniz Bandeira Duarte
{rubi, otto}@gta.ufrj.br

Grupo de Teleinformática e Automação - GTA
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica
P.O. Box 68504 - 21945-970 - Rio de Janeiro - RJ - Brazil
Tel.: + 55 21 2605010 - Fax: +55 21 2906626

Abstract

This paper aims to investigate mobile agent tradeoffs in network management by performing management task simulations. The mobile agent performs better than the SNMP when the number of managed network elements ranges between two limits: an inferior bound, associated with the number of messages that traverse backbone links, and a superior bound, related to the incremental size of the mobile agent, which turns migration difficult. This paper focuses on the performance analysis of two unload strategies to reduce the response time for the mobile agent: in the first approach, the mobile agent returns to the management station; in the second one, the mobile agent just sends data to the management station, both after visiting a fixed number of nodes. Results show that the response time decreases by 62, and 64%, for the first and second strategies, when compared with gathering all data without returning or sending these data to management station before finishing the task.

1 Introduction

Currently, most network management systems use SNMP (Simple Network Management Protocol) [1] and CMIP (Common Management Information Protocol) [2] protocols, which are based on a centralized paradigm. These protocols use the client-server model, on which the management station acts as a client that provides a user interface to the network manager and interacts with agents, which are servers that manage remote access to local information stored in a Management Information Base (MIB).

The operations available to the management station for obtaining access to the MIB are very low level. This fine grained client-server interaction, called micro-management, generates intense traffic and computational overload on the management station [3], resulting in scalability problems. At the same time that the networks grow, the computational power of the managed element increases, making possible to perform management functions in a distributed way [4]. In order to provide scalable solutions, the management system must be able to handle large numbers of elements without imposing performance problems or computer resource constraints. Moreover, the components of a management system must be able to monitor and gather data from the managed network with limited information loss and without placing too much demand on the elements of the network [5].

Network management can be distributed and scaled by the use of mobile agents, which are programs that help users to perform tasks on the network, acting on behalf of these users. These agents can move to the place where data are stored and select information the user wants by using intelligence; saving bandwidth, time, and money. Mobile agents decentralize processing and control, and, as a consequence, reduce the traffic around the management station, turn asynchronous agent-management communication (useful when there are unreliable or lossy links), distribute processing load, and increase the flexibility of the management agents' behavior.

Research activities related to mobile code in network management are recent [6]. Most researchers implement network management architectures using mobile agents. Management by Delegation (MbD) [7] has been the first paradigm to address decentralization and automation of management tasks by dynamically delegating management functions to agents. Pagurek et al. [8] implement a code mobility infrastructure and a suite of simple tools that interact with agents located on network components. El-Darieby and Bieszczcad [9] use this infrastructure to automate simple fault management tasks. Morin et al. [10] implement a management architecture that consists of a set of servers and managers communicating with SNMP agents located on the managed network elements. Geihs et al. [11] integrate SNMP and mobile agent paradigms to obtain a decentralized management architecture that is flexibly adjustable.

The performance of mobile agents in network management is also being investigated. El-Darieby and Bieszcad [9] and Geihs et al. [11] provide simple quantitative evaluation of mobile agent and client-server approaches. Baldi et al. [3] evaluate the tradeoffs of mobile code design paradigms in network management applications by developing a quantitative model that provides the bandwidth used by traditional and mobile code design of management functionalities. Gavalas et al. [12] analyze bandwidth utilization for the SNMP and two proposed polling modes. In the first approach, the network is partitioned into several domains and a single mobile agent is assigned to each of them; in the second polling scheme, the mobile agent is broadcasted to all managed elements and remains there for a number of polling intervals, before returning to the management station. Experimental results are analyzed in terms of bandwidth consumption and response time to obtain an aggregation of multiple variables (a health function). Sahai and Morin [10] perform measurements of bandwidth utilization of mobile agent and client-server applications on an Ethernet LAN. They also present a single case comparison of response time for the mobile agent and the client-server. Rubinstein and Duarte [13] simulate management tasks performed by mobile agents and SNMP ones, comparing both the approaches, on a topology that consists of a LAN of managed network elements connected to the management station by a bottleneck link.

Some preliminary results related to the gain due to the unload procedure of the mobile agent after visiting a fixed number of nodes are presented in [13]. This paper overcomes this unload-procedure analysis and evaluates mobile agent performance on 272-node-transit-stub topologies, which are similar in shape to the Internet ones. Some initial results related to the size of mobile agents on transit-stub topologies can be found in [14]. In this paper, a new strategy, sending data to the management station after visiting a fixed number of nodes, is evaluated, which performs better than the strategy in which the mobile agent returns to the management station.

This paper is organized as follows. Section 2, discusses the use of mobile agents in network management. Section 3 reports simulation results concerning the applicability of mobile agents in performing management tasks. At last, concluding remarks are presented in Section 4.

2 Network Management Using Mobile Agents

Mobile agents are created to perform their tasks in different network computers. A mobile agent can interrupt its execution and migrate from a machine to another, carrying data about agent's state that include information obtained from previous task executions. As the number of visited nodes grows, mobile agent size also increases, turning migration harder. One possible solution to this problem is to visit a fixed number of nodes, return or send all data to the agent home (reducing mobile agent size), and continue the task on the remaining nodes (Section 3.5). The initial size of a mobile agent also affects agent performance because the larger the size, the more difficult the migration (Section 3.3). This initial size depends on the task to be carried out and on the language used to implement the agent (Section 3.4).

In the SNMP, network management does not scale when the size or the complexity of the network increases because of the centralized processing and control. Mobile agents may solve this problem and improve management efficiency.

2.1 Simulation Model

The applicability of mobile agents in carrying out network management tasks is assessed by comparing mobile agent performance with the SNMP one.

The Network Simulator (NS) is used in this paper [15]. This discrete-event simulator provides several implemented protocols and mechanisms to simulate computer networks with node and link abstractions. Some UDP modules of the NS have had to be modified in order to effectively receive a packet, sending it to the next layer.

All simulations use transit-stub topologies, which are similar in shape to the Internet ones. The GT-ITM [16] generates random topologies, on which each node represents switches or routers and edges represent forwarding paths between switches (Figure 1). Each routing domain in the Internet can be classified as either a stub domain or a transit domain. A domain is a stub domain if the path connecting any two nodes u and v goes through that domain only if either u or v is in that domain. Transit domains do not have this restriction. The purpose of transit domains is to interconnect stub domains efficiently. A transit domain comprises a set of backbone nodes, which are typically fairly well connected to each other. In a transit domain, each backbone node also connects to a number of stub domains, via gateway nodes in the stubs.

The used management approach consider that the management station belongs to a node of a stub domain and managed network elements are located in other stub domains (Figure 1). Three different 272-node topologies are used. The management station controls groups of 16 network elements, which is the number of nodes of a stub domain.

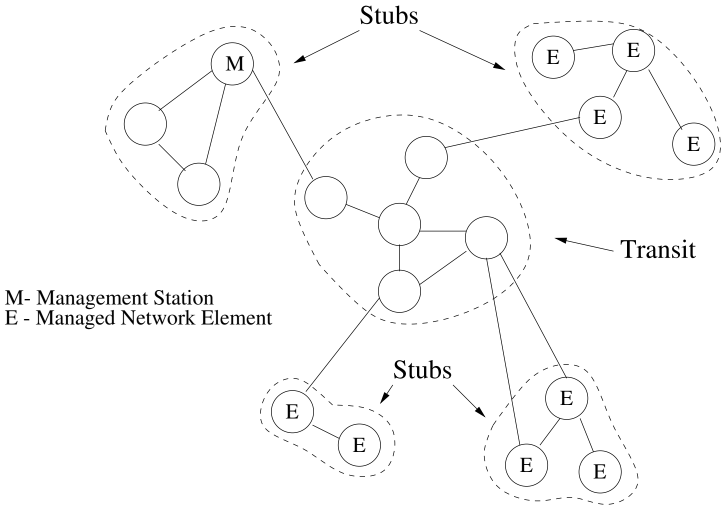

Figure 1: Network management on a transit-stub topology.

The simulation model assumes that links and nodes have no load, links are error-free, and processing time in application layer is not taken into account. UDP is used in all simulations and UDP and IP headers are considered (Figure 2). The Maximum Transmission Unit (MTU) used for all links is 1500 bytes, therefore, there is no fragmentation of SNMP messages since they are small. Every request/response of a variable is sent on a different message and the considered performance parameter is response time in retrieving the variables.

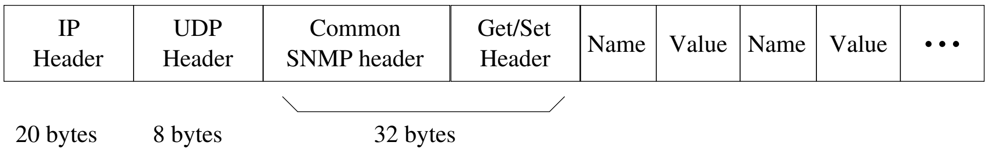

Figure 2: SNMP messages.

SNMP sends requests to all elements to be managed (one after receiving the response from the other); the mobile agent goes to an element to be managed, gathers the variable, and visits all other elements, from a stub domain to another. After finishing, the mobile agent returns to the management station.

3 Simulation Results

Various experiments are carried out in order to evaluate the effect on management performance of: number of managed network elements, data rate of the transit domain links, initial size of mobile agent, task to be performed, and unload strategies for the mobile agent.

3.1 Effect of the Number of Managed Network Elements

This section evaluates the effect of the number of managed network elements on response time. Each request/response is 50 bytes long. All links have a 2 Mbps data rate and latency links are about a few milliseconds. The initial size of the mobile agent is 5 kbytes.

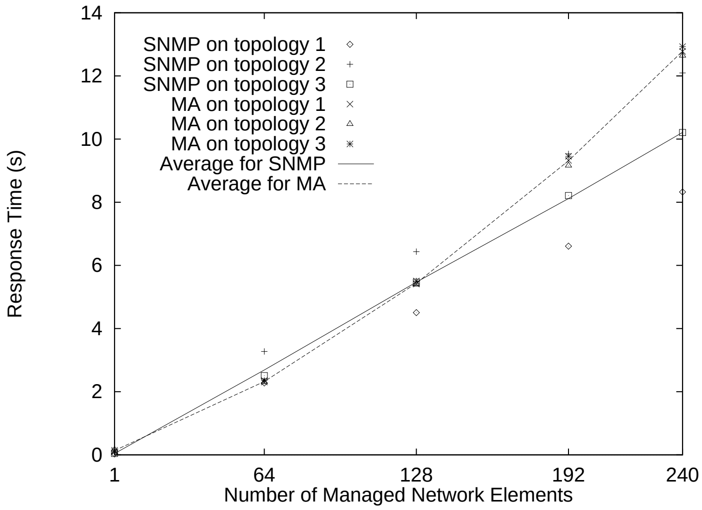

Figure 3: Response time per number of managed elements.

Figure 3 shows that the mobile agent's behavior does not change with the topology, but for the SNMP, the difference of performance among response times for the three topologies increases faster when the number of managed elements is large, due to the fact that SNMP packets traverse oftentimes the backbone links that depend on the topology. Figure 3 also presents the average response time per number of managed network elements. For a small number of managed elements, the SNMP performs better than the mobile agent due to the fact that an SNMP message is smaller than the initial size of the mobile agent. As the number of managed elements increases, response time for the SNMP grows proportionally, since the time to manage a stub domain is approximately the same for all stub domains. For the mobile agent, response time increases faster when the number of managed elements grows due to the incremental size of the mobile agent. Therefore, the mobile agent performs better that the SNMP when the number of managed network elements ranges between two limits.

3.2 Effect of the Data Rate of the Transit Domain Links

This section evaluates the effect of the data rate of the transit domain links. Each request/response is 50 bytes long. Three different data rates are used (0.064, 1, and 2 Mbps) and latency links are the same of the previous section. The initial size of the mobile agent is 5 kbytes.

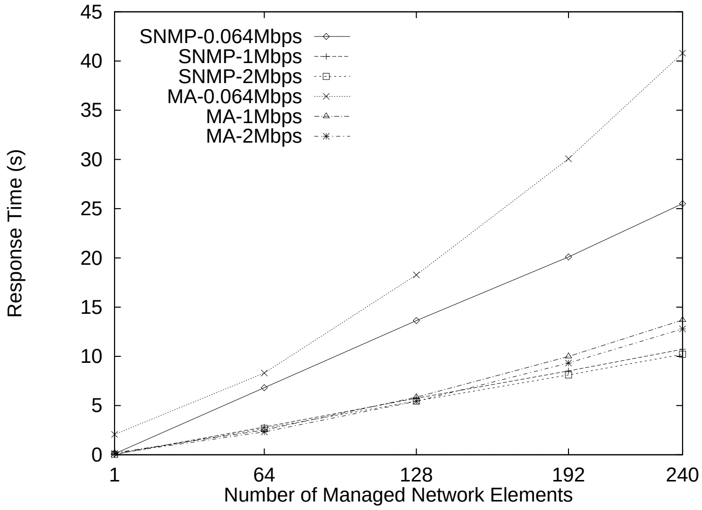

Figure 4: Response time for different data rates of the transit domain links.

In Figure 4, for the smaller data rate, the SNMP and the mobile agent present the higher response time since the small data rate turns difficult the transference of the mobile agent and SNMP packets. For the other values for the data rate, the mobile agent and the SNMP's behavior is the same described in the previous section.

3.3 Effect of the Initial Size of Mobile Agent

The effect of the initial size of mobile agent is also analyzed. Three different initial sizes of mobile agent are used (1, 5, and 9 kbytes). The topologies and the task are the same of Section 3.1.

As the number of managed network elements increases, the performance difference among the various mobile agent sizes increases faster (Figure 5). For example, when the number of managed network elements doubles from 64 to 128, the response time for mobile agent sizes 1, 5, and 9kB goes from 1.27023 to 3.36017 s, from 2.3209 to 5.43483 s, and from 3.3609 to 7.49883 s, respectively. Therefore, the performance difference among the three mobile agent sizes for these numbers of managed network elements raises from approximately 1 to 2 s.

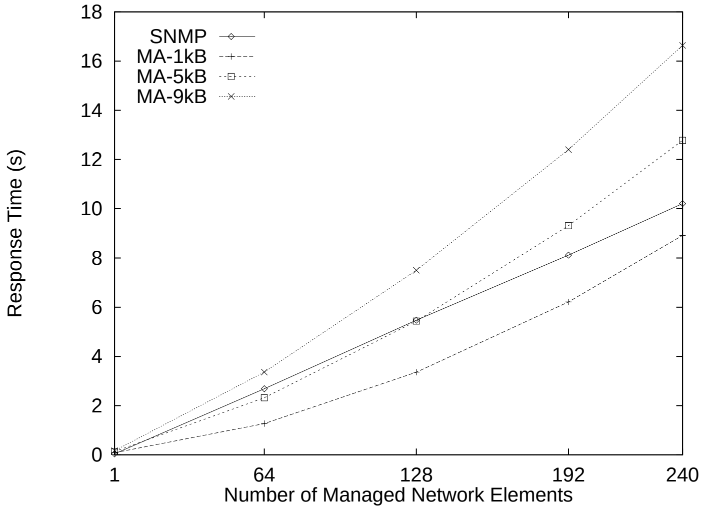

Figure 5: Response time for different mobile agent sizes.

3.4 Effect of the Task to Be Performed

An analysis related to mobile agent and SNMP's behavior for different tasks is performed. Three tasks are used: task t_1 is the same task used in previous sections, task t_2 has a 100-byte-long request/response, and task t_3 is associated with a 150-byte-long variable. The topologies and the mobile agent size are the same of the previous section.

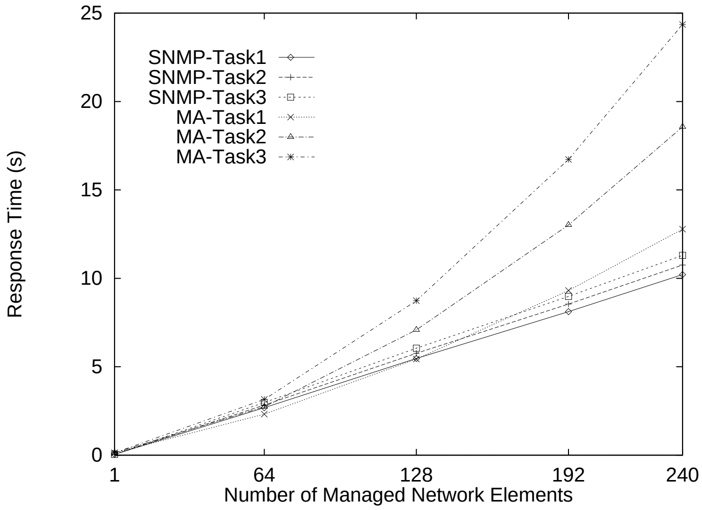

Figure 6: Response time for different tasks.

As depicted in Figure 6, response time differs when using the mobile agent or the SNMP in network management tasks. For the SNMP, the task does not have a great influence due to the fact that the number of bytes exchanged between the management station and the managed network elements is small; but for the mobile agent, as the number of bytes exchanged increases (e.g., from task t_1 to task t_3), the response time also increases since the mobile agent will move with higher difficulty.

3.5 Effect of Unload Strategies

It has been shown in the previous sections that the mobile agent size increases with the number of visited nodes and, as a consequence, the migration becomes difficult. This section evaluates the performance improvement of two unload strategies: in the first one, s_1, the mobile agent returns to the management station, and for s_2, the mobile agent just sends the results to the management station. The simulations consider that the mobile agent visits a fixed number of nodes "per trip", i.e., the number of visited network elements before returning or just sending the results to the management station. The considered tasks are the three previous one but their purpose is to retrieve the variables from all the 240-managed-network elements. The topologies and the mobile agent size are the same of the previous experiment. The number of visited nodes per trip varies from 1 to 240.

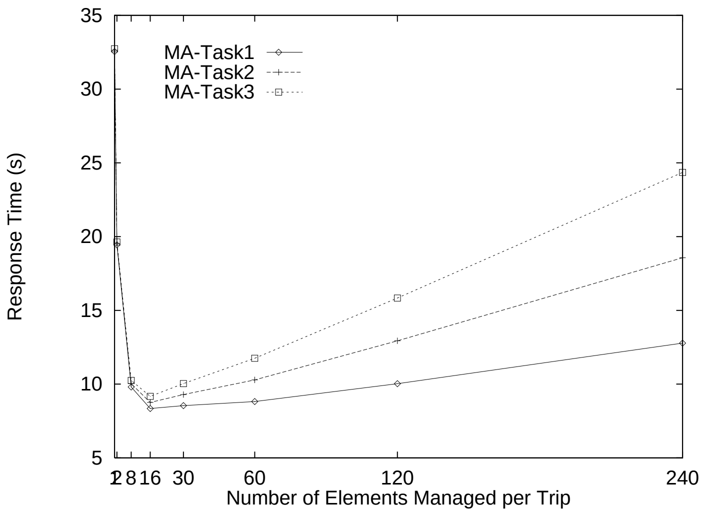

Figure 7: Response time for strategy s_1 and different tasks.

Table 1: Response time for strategy s_1 and different tasks.

Elements per trip	Task t_1 (s)	Task t_2 (s)	Task t_3 (s)
1	32.5431	32.6391	32.7351
2	19.4405	19.5365	19.6325
8	9.80777	10.0298	10.2518
16	8.34643	8.75743	9.16843
30	8.54463	9.29023	10.0358
60	8.82087	10.2857	11.7505
120	10.0303	12.9347	15.8391
240	12.7821	18.5663	24.3505
SNMP	10.2102	10.7558	11.3014

Figure 7 shows that the response time for strategy s_1 decreases sharply when the number of elements managed per trip is small since the agent visits few nodes and returns to the management station. As the number of nodes visited per trip goes on raising, response time decreases up to a point when it starts to increase due to agent migration difficulty related to agent size. An "optimum point" for this strategy is associated with visiting 16 nodes and then returning to the management station to unload (Table 1). According to this Table, the response time decreases by 35, 53, and 62%, for tasks t_1, t_2, and t_3, when comparing each optimum point with gathering all 240 variables in one trip. This optimized strategy performs better than the SNMP for all tasks (Table 1). For a small number of elements per trip, the task size does not have a great influence since the mobile agent size is small, but when the number of elements per trip is large, e.g., 240, the difference among the tasks is significant. Thus, the greater the task size the greater the improvement of this strategy.

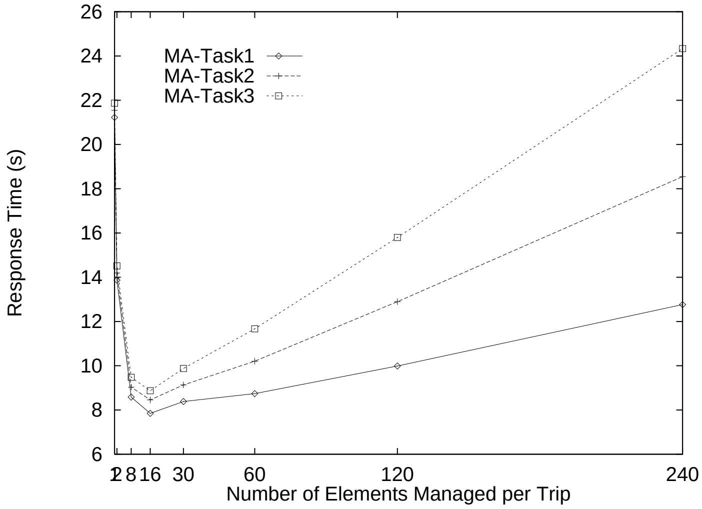

Figure 8: Response time for strategy s_2 and different tasks.

Table 2: Response time for strategy s_2 and different tasks.

Elements per Trip	Task t_1 (s)	Task t_2 (s)	Task t_3 (s)
1	21.2239	21.5447	21.8655
2	13.8691	14.1916	14.5141
8	8.5815	9.03123	9.48097
16	7.84857	8.45743	8.86843
30	8.38463	9.13023	9.87583
60	8.74087	10.2057	11.6705
120	9.9903	12.8947	15.7991
240	12.7621	18.5463	24.3305
SNMP	10.2102	10.7558	11.3014

Figure 8 shows that the response time for the second strategy also has an optimum point. According to Table 2, the response time decreases by 39, 54, and 64%, for tasks t_1, t_2, and t_3, when comparing each optimum point with gathering all 240 variables in one trip.

Comparing both the strategies for task t_1, strategy s_2 performs better when the number of elements managed per trip is small (Figure 9), since the mobile agent's code is not sent to the management station. As this number of elements per trip increases, both the approaches

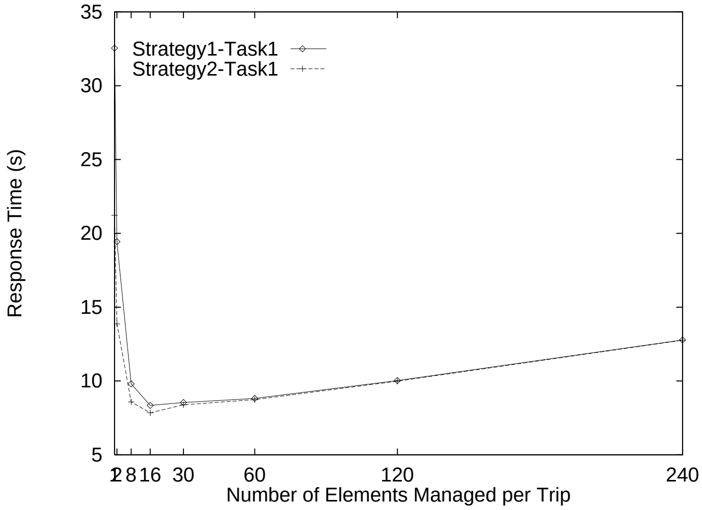

Figure 9: Response time for different tasks.

behave almost the same way, due to the fact that the size of data gathered by the agent is much greater than the size of the mobile agent's code.

The effect of the initial size of the mobile agent is also analyzed. The considered task is t_1 and all other parameters are the same of the previous experiment.

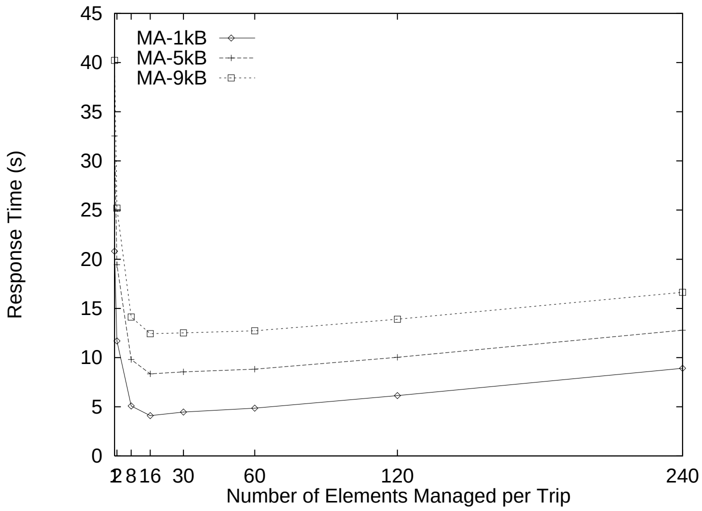

Figure 10: Response time for strategy s_1 and different initial sizes.

Figure 10 shows that all three different initial sizes have an optimum point. In Table 3, the response time decreases by 54, 35, and 25%, for sizes 1, 5, and 9 kbytes, when comparing each optimum point with gathering all 240 variables in one trip. The smaller the initial size the greater the gain of this strategy. When sending data to the management station (Figure 11 and Table 4), the response time decreases by 57, 39, and 30%, for sizes 1, 5, and 9 kbytes. These optimized strategies also perform better than the SNMP (Tables 3 and 4) when the initial size is 1 or 5 kbytes.

Table 3: Response time for strategy s_1 and different initial sizes.

Elements per Trip	1 kbyte	5 kbytes	9 kbytes
1	20.8167	32.5431	40.2231
2	11.6932	19.4405	25.2005
8	5.08457	9.80777	14.1278
16	4.09717	8.34643	12.4264
30	4.46357	8.54463	12.5126
60	4.8596	8.82087	12.7249
120	6.13223	10.0303	13.9023
240	8.9154	12.7821	16.6381
SNMP	10.2102	10.2102	10.2102

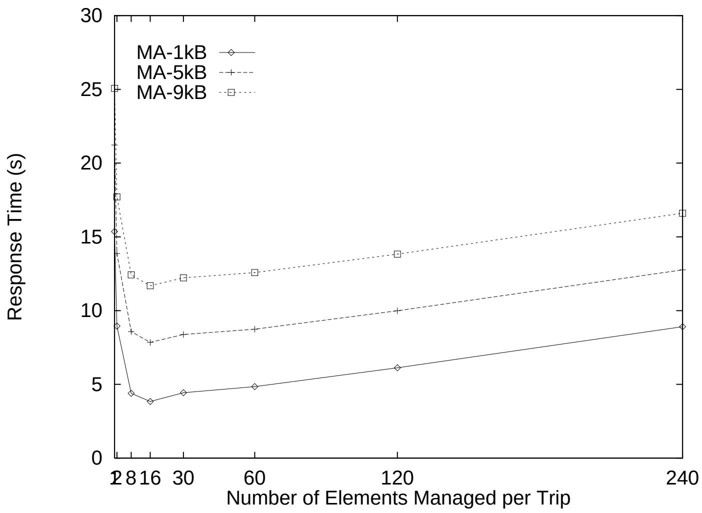

Figure 11: Response time for strategy s_2 and different initial sizes.

Table 4: Response time for strategy s_2 and different initial sizes.

Elements per Trip	1 kbyte	5 kbytes	9 kbytes
1	15.3607	21.2239	25.0639
2	8.94787	13.8691	17.7091
8	4.39523	8.5815	12.4215
16	3.8393	7.84857	11.6886
30	4.43157	8.38463	12.2246
60	4.8436	8.74087	12.5809
120	6.12423	9.9903	13.8303
240	8.9114	12.7621	16.6021
SNMP	10.2102	10.2102	10.2102

4 Conclusion

This paper has analyzed tradeoffs of mobile agents in network management tasks. The performance of mobile agents has been compared with the SNMP one in several simulations.

Response time results show that the mobile agent is more influenced by the task to be performed, since the number of bytes of an SNMP message, exchanged between the management station and the managed network element, is small. The results also show that the mobile agent performs better than the SNMP when the number of managed network elements ranges between two limits: an inferior bound, associated with the number of messages that pass through backbone links, and a superior bound, related to incremental size of mobile agent, which turns migration difficult.

Moreover, the mobile agent performance increases when the agent returns/sends data to the management station (respectively strategies s_1 and s_2), after visiting a fixed number of nodes. Response time decreases by 35, 53, and 62%, for tasks t_1, t_2, and t_3, when compared with gathering all variables without returning to the management station before finishing the task. Using strategy s_2, response time reduces by 39, 54, and 64%, for tasks t_1, t_2, and t_3. For both the strategies, the response time of the mobile agent is smaller than the SNMP one.

For the strategy s_1, the response time reduces by 54, 35, and 25%, for initial sizes 1, 5, and 9 kbytes, when comparing each optimum point with gathering all the 240 variables in one trip. When sending data to the management station, the response time decreases by 57, 39, and 30%, for sizes 1, 5, and 9 kbytes. Optimized strategies s_1 and s_2 also perform better than the SNMP when the initial size is 1 or 5 kbytes.

Network management using mobile agents seems to be a good approach and the response time is made lower by the presented strategies.

References

[1] W. Stallings, "SNMP and SNMPv2: The infrastructure for network management", *IEEE Communications Magazine*, vol. 36, no. 3, pp. 37–43, Mar. 1998.

[2] Y. Yemini, "The OSI network management model", *IEEE Communications Magazine*, vol. 31, no. 5, pp. 20–29, May 1993.

[3] M. Baldi and G. P. Picco, "Evaluating the tradeoffs of mobile code design paradigms in network management applications", in *Proceedings of the 20th International Conference on Software Engineering (ICSE'98)*, Kyoto, Japan, pp. 146–155, Apr. 1998.

[4] K. Meyer, M. Erlinger, J. Betser, C. Sunshine, G. Goldszmidt and Y. Yemini, "Decentralizing control and intelligence in network management", in *Proceedings of the 4th International Symposium on Integrated Network Management*, Santa Barbara, USA, May 1995.

[5] P. Haggerty and K. Seetharaman, "The benefits of CORBA-based network management", *Communications of the ACM*, vol. 41, no. 10, pp. 73–79, Oct. 1998.

[6] V. A. Pham and A. Karmouch, "Mobile software agents: An overview", *IEEE Communications Magazine*, vol. 36, no. 7, pp. 26–37, July 1998.

[7] G. Goldszmidt and Y. Yemini, "Delegated agents for network management", *IEEE Communications Magazine*, vol. 36, no. 3, pp. 66–70, Mar. 1998.

[8] A. Bieszczad, B. Pagurek and T. White, "Mobile agents for network management", *IEEE Communications Surveys*, vol. 1, no. 1, Sept. 1998.

[9] M. El-Darieby and A. Bieszczad, "Intelligent mobile agents: Towards network fault management automation", in *Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, USA, pp. 610–622, May 1999.

[10] A. Sahai and C. Morin, "Towards distributed and dynamic network management", in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'98)*, New Orleans, USA, Feb. 1998.

[11] M. Zapf, K. Herrmann and K. Geihs, "Decentralized SNMP management with mobile agents", in *Sixth IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, USA, pp. 623–635, May 1999.

[12] D. Gavalas, D. Greenwood, M. Ghanbari and M. O'Mahony, "Using mobile agents for distributed network performance management", in *Proceedings of the 3rd International Workshop on Intelligent Agents for Telecommunication Applications (IATA'99)*, Stockholm, Sweden, Aug. 1999.

[13] M. G. Rubinstein and O. C. M. B. Duarte, "Evaluating tradeoffs of mobile agents in network management", *Networking and Information Systems Journal*, vol. 2, no. 2, July 1999.

[14] M. G. Rubinstein and O. C. M. B. Duarte, "Evaluating the performance of mobile agents in network management", in *Proceedings of the IEEE Global Telecommunications Conference GLOBECOM'99 (to be published)*, Rio de Janeiro, Brazil, Dec. 1999.

[15] K. Fall and K. Varadhan, "NS Notes and Documentation", tech. rep., The VINT Project, Jan. 1999.

[16] E. W. Zegura, K. L. Calvert and M. J. Donahoo, "A quantitative comparison of graph-based models for internet topology", *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 770–783, Dec. 1997.