Analyzing Mobile Agent Scalability in Network Management*

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

Abstract

This paper aims to investigate mobile agent scalability in network management, in order to find when mobile agents can improve the management efficiency. Mobile agent performance is compared with the SNMP one in management task simulations. Response time results show that the mobile agent is less sensitive to the latency and the bandwidth of a bottleneck link that connects the management station to a remote LAN, but the mobile agent is more influenced by the task to be performed. The results also show that the mobile agent performs better when the number of managed network elements ranges between two limits: an inferior bound, associated with the number of messages that traverse the bottleneck link, and a superior bound, related to the incremental size of the mobile agent, which turns migration difficult. Moreover, the performance increases when the agent returns to the management station after visiting a fixed number of nodes.

1 Introduction

Agents are programs that help users to perform tasks, acting on behalf of these users. If these agents can execute their tasks on the network, they are called mobile agents. Mobile agents can move to the place where data are stored and select information the user wants by using intelligence; saving bandwidth, time, and money.

Network management can benefit from the use of mobile agents, due to the fact that, in a general manner, this management is based on a centralized paradigm. SNMP (Simple Network Management Protocol) [1] and CMIP (Common Management Information Protocol) [2] use the

*This paper has been supported by UFRJ, FUJB, CNPq, CAPES, COFECUB, and REENGE.

client-server model, on which the manager centralizes information and gives orders to perform repair actions, and the agent collects and saves information in a database and executes manager orders. This centralized approach does not scale when the size or the complexity of the network increases and may not work well when there is a point of failure on the network or in the manager. Mobile agents decentralize processing and control, and, as a consequence, reduce the traffic around the management station, turn asynchronous agent-manager 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 still recent [3]. Management by Delegation (MbD) [4] has been the first paradigm to address decentralization and automation of management tasks by dynamically delegating management functions to agents. Mobile agent applications in the OSI functional areas of network management are being developed by Pagurek et al. [5]. They have implemented a code mobility infrastructure and a suite of simple tools that interact with agents located on network components. Baldi et al. [6] evaluate the tradeoffs of mobile code design paradigms in network management applications by developing a quantitative model for traditional and mobile code design of management functionalities. Morin et al. [7] implement a management architecture that consists of a server and several managers communicating with SNMP agents located on the managed network elements. They perform measurements of bandwidth utilization for this architecture on an Ethernet LAN, while Rubinstein and Duarte [8] simulate management tasks performed by mobile agents and SNMP ones, comparing both the approaches, on topologies similar in shape to the Internet.

This paper aims to investigate mobile agent scalability in network management. We analyze the performance of a management system and compare mobile agent and SNMP approaches. The performance is evaluated for: the latency and the data rate of the link that connects the management station to the managed LAN, the initial size of mobile agent, the task to be performed, and the gain due to the unload procedure of the mobile agent after visiting a fixed number of nodes.

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

2 Network Management Systems

Currently, most network management systems are centralized, but some steps towards decentralization have already been taken and new decentralized approaches are also being developed [9].

SNMP and CMIP protocols, proposed by the Internet Engineering Task Force (IETF) and the International Organization for Standardization (ISO), are based on 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 their local information.

IETF and ISO have taken some steps towards decentralization. In event notification, SNMP agents notify the management station upon the occurrence of a few significant events by using traps, i.e., messages sent without an explicit request from the management station. The ISO approach uses more complex agents that have higher processing capacity. In both cases, the agent is only responsible for the notification of an event. SNMPv2 adopts a decentralized approach where there are multiple top-level management stations, the management servers. Each such server is responsible for managing agents, but it can delegate responsibility to an intermediate manager. The intermediate manager, also called proxy agent, plays the role of a manager, in order to monitor and control the agents under its responsibility; and it also plays the role of an agent to provide information and accept control from a higher-level management server. SNMPv3 makes additional minor functional changes to the SNMPv2 and incorporates a new security approach [1]. IETF has also proposed RMON (Remote MONitoring) [10] that uses network monitoring devices called monitors or probes. By monitoring packet traffic and analyzing the headers, the probes provide information about links, connections among stations, traffic patterns, and status of network nodes. A probe can detect failures, misbehaviors, and identify complex events even when not in contact with the management station.

New decentralized architectures are also being developed and some of them use mobile agents to decentralize processing and control from the management station [8].

2.1 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 3.4). The initial size of a mobile agent also affects agent performance because the larger the size, the more difficult the migration (Section 3.2). This initial size depends on the task to be carried out (Section 3.3) and on the language used to implement it.

Lots of advantages [11] may justify agent utilization in network management, such as:

- Reduced cost: since management functions require the transfer of a great volume of data on the network, it may be better to send an agent to perform the task directly on management agents, where data are stored. Thus, the agent can filter and select just the relevant information and transmit them to the management station.
- Asynchronous processing: an agent can be sent through the network to perform tasks on other nodes, and while the agent is out of its home node, this node can be out of operation.
- Distributed processing: low capacity computers can be efficiently used in order to perform simple management tasks, distributing processing previously concentrated on the management station.

- Flexibility: a new behavior for the management agents can be determined by the management station by sending a mobile agent with a new execution code, substituting in real-time the old one.

For 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 can be used in order to solve this problem and it is important to investigate when they improve the management efficiency.

3 Simulation Results

The applicability of mobile agents in carrying out network management tasks is assessed by comparing the mobile agent performance with the SNMP one. The topology used in these simulations consists of a LAN of Managed Network Elements (MNEs) connected to the management station by a bottleneck link called l (Figure 1). For the Ethernet LAN, the number of nodes is 250, the data rate is 10 Mbps, and the latency is 10 μs. The considered performance parameters are bandwidth consumption on link l, evaluated by the number of bytes transmitted and received by a management station, and response time in retrieving SNMP variables (MIB-II objects [12]).

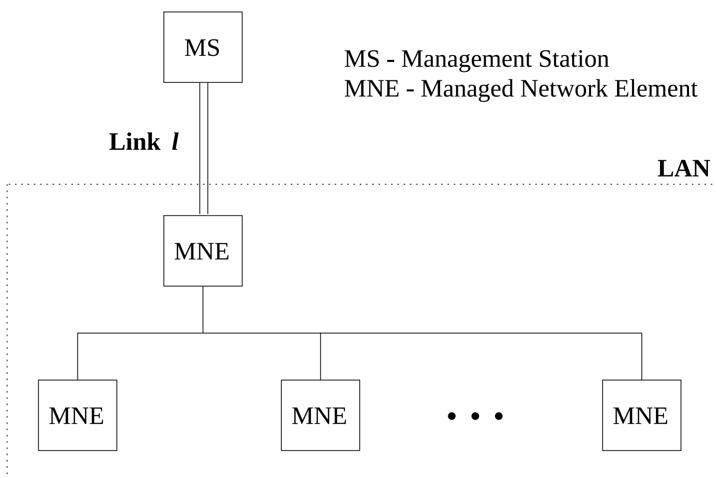

Figure 1: Topology used in all simulations.

The Network Simulator (NS) [13] is used in this paper. This discrete-event simulator provides several implemented protocols and mechanisms to simulate computer networks with node and link abstractions. In these simulations, we have used UDP and Ethernet implemented functions, but some UDP modules of the NS have had to be modified in order to effectively receive a packet, sending it to the next layer.

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 also considered. 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.

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. After finishing, the mobile agent returns to the management station.

Various experiments are carried out in order to evaluate the effect on management performance of: the latency and the data rate of the link l, the initial size of mobile agent, the task to be performed, and the gain due to the unload procedure of the mobile agent after visiting a fixed number of nodes.

3.1 Effect of the Latency and Data Rate of the Link l

In this section, we evaluate the effect of the latency and data rate of the link l, when retrieving an SNMP variable named *ifInErrors*, which denotes the number of received packets discarded because of errors, from interfaces of LAN stations. Three different values for the latency and the data rate of the link l are used, as in Table 1, and the mobile agent size is 5 kbytes.

Table 1: Values for the latency and the data rate of the link l.

Link l Values	Latency (ms)	Data Rate (Mbps)
v_1	10	0.64
v_2	4	2
v_3	0.1	10

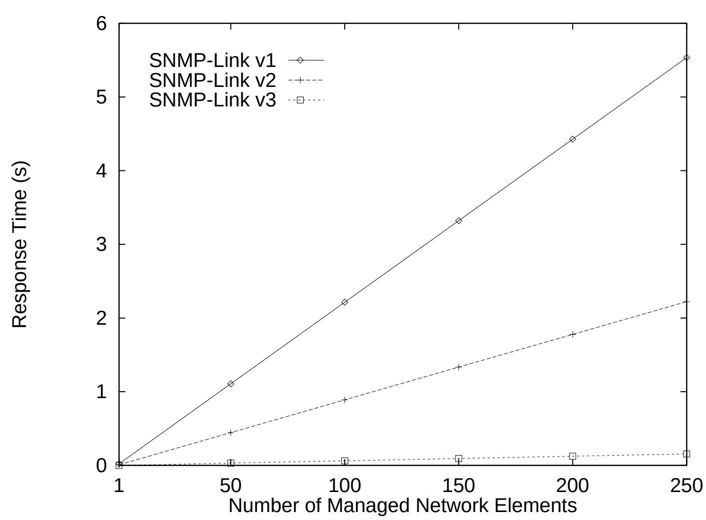

Figure 2: Response time for the SNMP.

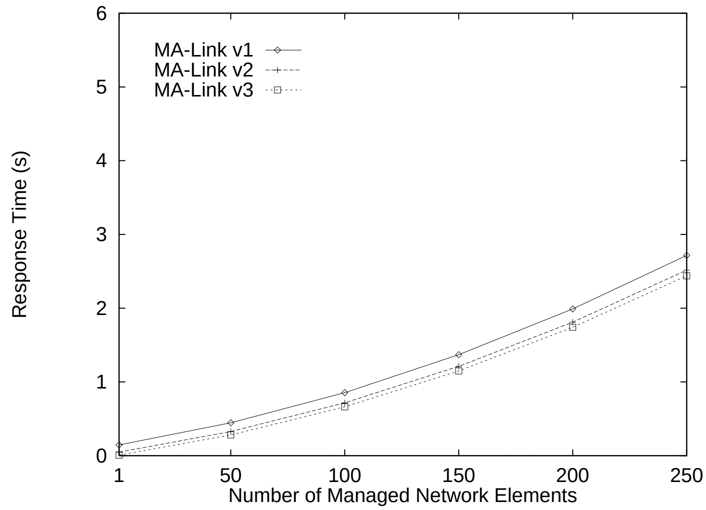

Figure 3: Response time for the mobile agent.

Figure 3 shows that mobile agent's behavior does not change significantly with the topology, but for the SNMP, the difference of performance among response times for the three topologies increases faster when the number of MNEs is large (Figure 2), due to the fact that SNMP packets traverse oftentimes the link l. Response time for the SNMP grows approximately linearly with the number of MNEs. For the mobile agent, the response time increases faster when the number of MNEs grows, due to the incremental size of the mobile agent. For the link v_3 (Table 1), the SNMP performs better than the mobile agent mainly because the link latency is small, but for the link v_1, the mobile agent has a smaller response time since the link latency is large. If the latency is medium (link v_2), the mobile agent performs better when the number of MNEs ranges between two limits (Section 3.2).

3.2 Effect of the Initial Size of Mobile Agent

The effect of the initial size of mobile agent is also analyzed for a link l value equal to v_2 (Table 1). Five different initial sizes of mobile agent are used (1, 3, 5, 7, and 9 kbytes). The task is the same of the previous experiment.

For a small number of MNEs, the SNMP uses fewer bytes than the mobile agent to perform the task, but as the number of MNEs increases, the overhead associated with several retrievals (PDU GetRequest, UDP, and IP's headers) turns mobile agent utilization more suitable (Figure 4). As mobile agent size increases, the number of bytes associated with the management station increases proportionally. For the response time, as the number of MNEs increases, the performance difference among the various mobile agent sizes increases faster (Figure 5). Analyzing the curves MA-5kB and SNMP, for a small number of MNEs, the mobile agent performs better, but when the number of MNEs is greater than 200, SNMP performance is better since the mobile agent size is significant.

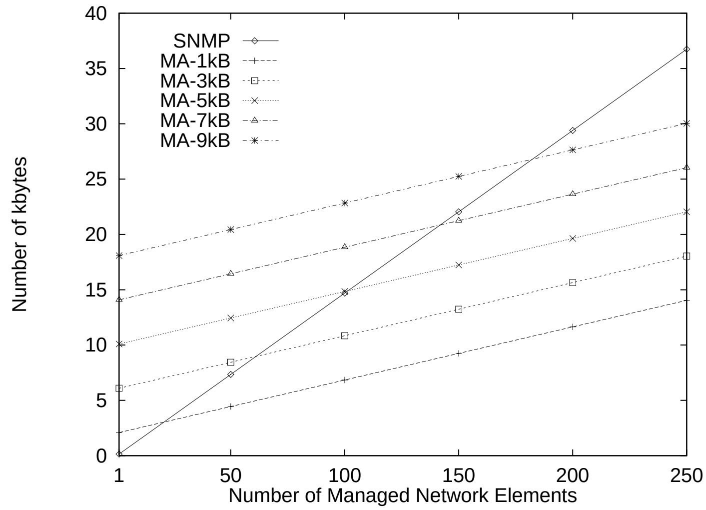

Figure 4: Number of bytes for different mobile agent sizes.

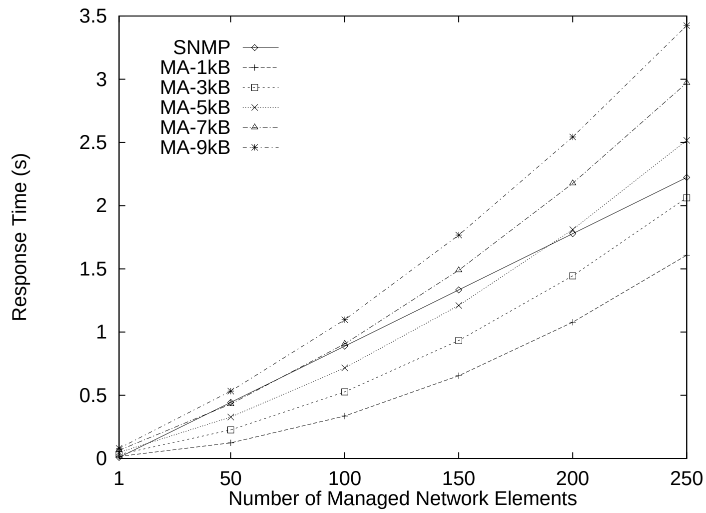

Figure 5: Response time for different mobile agent sizes.

3.3 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 is related to the retrieval of the SNMP variable named *sysContact*, which is the name of contact person and how to contact them; and task t_3 is related to the SNMP variable called *sysDescr*, which is a textual description of the system. Request and response packet sizes for the tasks, including all overhead (PDU GetRequest/GetResponse, UDP, and IP headers) are in Table 2. The link l value is the same of the previous experiment, and the mobile agent size is 5 kbytes.

Figure 6 shows that the number of bytes varies with the task to be performed because the greater the number of bytes of every request/response, the larger the total number of bytes transmitted

Table 2: Request and response packet sizes for the tasks.

Task	Request (bytes)	Response (bytes)
t_1	71	76
t_2	69	91
t_3	66	173

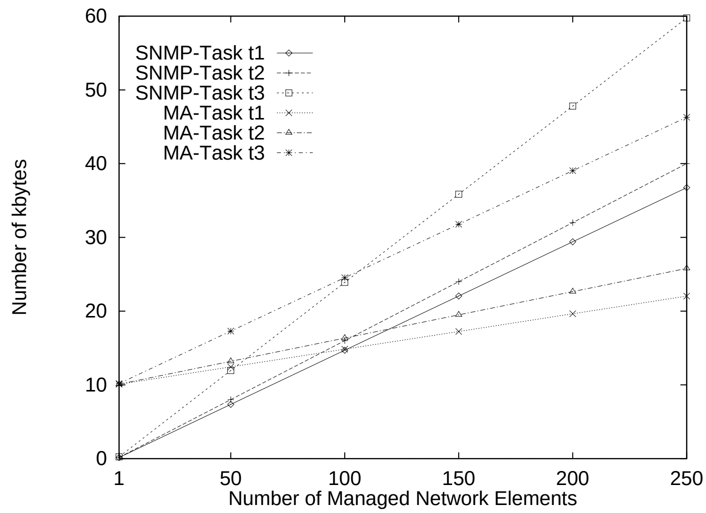

Figure 6: Number of bytes for different tasks.

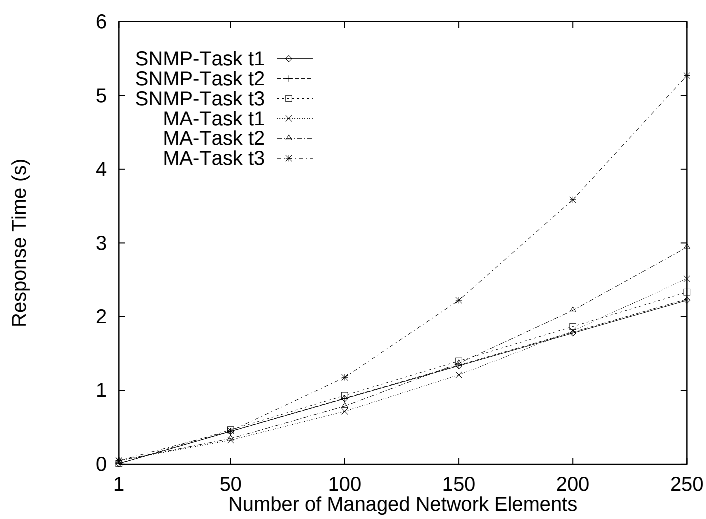

Figure 7: Response time for different tasks.

and received by the management station. All mobile agent curves start at 10 kbytes since agents traverse the bottleneck link two times and their initial size is 5 kbytes. In Figure 7, response time varies in a different way when using the mobile agent or the SNMP. 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 MNEs is small; but for the mobile agent, as the number of exchanged bytes 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.

3.4 Effect of Returning to the Management Station

It has been shown in the previous sections that the mobile agent size increases with the number of visited nodes and, as a consequence, migration becomes difficult. In this section, we evaluate the performance gain of the strategy of returning to the management station when the mobile agent size becomes large. The simulations consider that the mobile agent visits a fixed "number of nodes per trip", i.e., the number of visited MNEs before returning to the management station to "unload". The considered tasks are the three previous ones but their purpose is to retrieve the variables from all 250 MNEs. The link l value and the mobile agent size are the same of the previous experiment. The number of visited nodes per trip varies from 1 to 250.

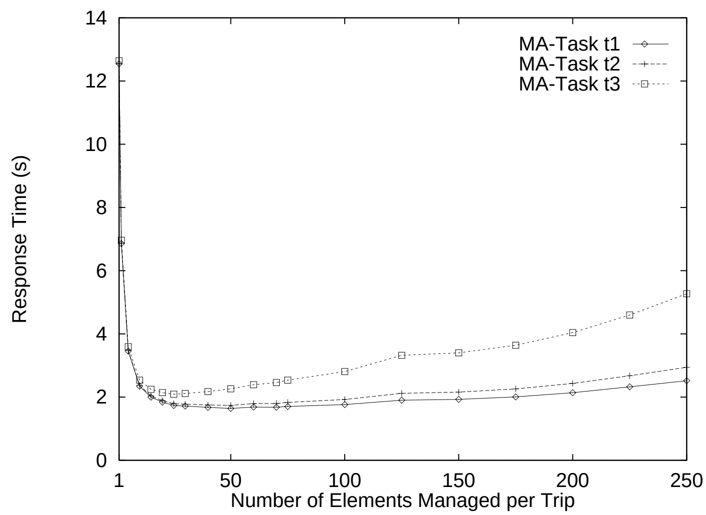

Figure 8: Response time when returning to the management station.

Figure 8 shows that the response time 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 increasing, response time decreases up to a point when it starts to increase due to agent migration difficulty related to the agent size. The "optimum point" varies with the task; for tasks t_1 and t_2, visiting 50 nodes and then returning to the management station provides the best result, while for task t_3, 25 is the number of nodes to be visited per trip. For the number of bytes, it decreases when the number of elements managed per trip increases, as in Figure 9, because the more the mobile agent returns to the management station, the larger the number of bytes transmitted and received by this station.

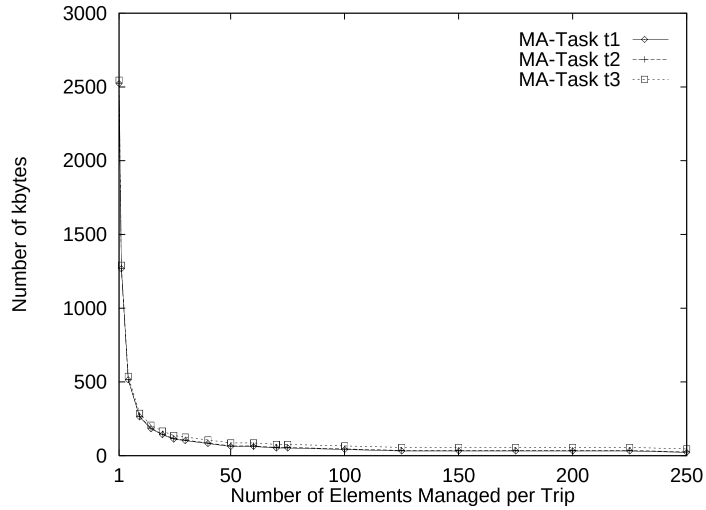

Figure 9: Number of bytes when returning to the management station.

4 Conclusion

This paper has analyzed the scalability 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 less sensitive to the latency and the bandwidth of the bottleneck link that connects the management station to the remote LAN, due to the fact that SNMP packets traverse oftentimes the bottleneck link, but it is more influenced by the task to be performed, since, for the SNMP, the number of bytes exchanged between the management station and the managed network elements 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 traverse the bottleneck link, and a superior bound, related to the incremental size of mobile agent, which turns migration difficult. Moreover, the mobile agent performance increases when the agent returns to the management station, after visiting a fixed number of nodes. For the number of bytes, the mobile agent performs better than the SNMP when the number of managed network elements exceeds a value related to the overhead of several SNMP retrievals.

We conclude that the mobile agent paradigm significantly improves the network management performance when most of the agent movement is inside high bandwidth LANs; in other words, when only a small percentage of the agent movement concerns the management station and wide area links that are bottlenecks.

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

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

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

[6] 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.

[7] 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.

[8] 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*, Rio de Janeiro, Brazil, Dec. 1999.

[9] M. Kahani and H. W. P. Beadle, "Decentralised approaches for network management", *Computer Communications Review*, vol. 27, no. 3, pp. 36–47, July 1997.

[10] S. Waldbusser, "Remote network monitoring management information base". RFC 1757, Feb. 1995.

[11] H. S. Nwana, "Software agents: an overview", *Knowledge Engineering Review*, vol. 11, no. 3, pp. 1–40, Sept. 1996.

[12] K. McCloghrie and M. Rose, "Management information base for network management of TCP/IP-based internets: MIB-II". RFC 1213, Mar. 1991.

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