Distributed Retrieval of Management Information: Is It About Mobility, Locality or Distribution?

P. Simões, J. Rodrigues, L. Silva, F. Boavida
CISUC - University of Coimbra
Pólo II, Dep. Eng. Informática
3030 Coimbra – Portugal
{psimoes, jpr, luis, boavida}@dei.uc.pt

Abstract

In this paper we present results from an experimental study addressing the use of mobile agents in the retrieval of management information. We compare several agent-based models for distributed collection and processing of management data, focusing on performance but also considering network traffic and setup costs. This study reveals that in many situations the performance of mobile agent systems is mainly determined by distribution, rather than locality. Furthermore, it shows that despite its importance in the flexibility and adaptability of the management system, agent mobility does not increase the system performance.

Keywords

Mobile Agent-based Management, Performance of Distributed Monitoring

1. Introduction

Distributed network management (DNM) is currently a broad designation for a large number of distinct approaches to decentralized management [1]. The common link between these approaches is the notion that distribution of the management process over the network will result in better management solutions, characterized by increased levels of flexibility, efficiency and robustness. Early work on MbD [2] triggered this quest for decentralization, and was later followed by several proposals. Some of them focused on the evolution in the context of classic network management architectures [3-4] while others, on the other hand, were more closely related to the advances in the field of distributed computing: CORBA-based management [5], Intelligent Agent-based management [6], Java-based management [7], Jini-based management [8], etc.

Mobile agents (MA) represent one of the last paradigms going over this cycle of conversion to DNM. The main difference between mobile agents and other distributed computing paradigms is the ability to dynamically change the location of the program in the middle of its execution, without loosing its execution state.

This potentially results, among other advantages, in more natural programming metaphors and better adaptation to mobile and disconnected computing [9].

Over the last few years several R&D projects assessed the interest of mobile agent technology (MAT) in the field of DNM. Analytical studies [10-13] addressed issues like efficiency, scalability, sensibility to bandwidth constraints, and size and complexity of mobile agents. However, despite their quality and completeness, the relation between these models and real world applications is still not clear. Experimental evaluation was also addressed by several projects. Some of them focused on the development of specific management frameworks and their experimental data is used mainly for validation purposes [14-16]. Other studies, however, provide a more generic perspective. Bohoris et al. complemented previous analytical work [12] with an experimental comparison between several distributed computing technologies [17]. Lipperts proposed utility functions to determine whether it is more efficient to conduct remote client/server transactions or to send the client (i.e. the mobile agent) to the managed node in order to perform these transactions locally [18]. Gavalas et al. focused on the retrieval of large volumes of management information, achieving significant performance and network traffic improvements with the usage of mobile agents to collect and process this information directly at the source. Several distribution strategies are combined with semantic compression of the management information [19], table filtering [20] and more scalable hierarchic distribution schemes [21]. The main interest of this line of work is the exploration and refinement of data compression techniques, although the studies about the best strategies to distribute the mobile agents through the network are also quite interesting. However, the small test beds in use compromise extrapolation of the conclusions to larger systems.

Our line of work complements these studies and is also based on the scenario of cyclic retrieval of management information from a set of managed nodes. In each cycle, designated as a poll, several SNMP transactions are conducted with each node, either locally or remotely. The management information is then processed, compressed and sent to the central management station. The use of a larger set of managed nodes and the introduction of static delegation, as well as the focus on the steady-state behavior of the system (isolating setup costs from running costs) provided a somehow different view on this subject.

The rest of the paper is organized as follows: Section 2 describes the experimental test bed, while Section 3 and Section 4 address the performance of constrained-mobility and mobility-based models. Section 5 presents network traffic measurements and Section 6 analyses the effect of network bandwidth. Section 7 discusses setup costs and Section 8 concludes the paper.

2. Conditions for the Experimental Evaluation

Five different distribution models were considered (Figure 1). The **static centralized model** (SC) corresponds to the classical SNMP-based management applications where a central node directly interacts with each managed node. In the **migratory model** (MG) a single mobile agent visits each node in order to perform local SNMP transactions, and retrieved information is then compressed and carried by the agent until it reaches the last managed node. When this finally happens, the agent sends the aggregated data to the central node. In the **migratory delegated model** (MD) this process is distributed by several mobile agents that work in parallel. In the **master/worker model** (MW) a mobile agent is sent to each managed node and stays there, performing local SNMP transactions, compressing the retrieved management data and sending it to the central node. In the **static delegated model** (SD) the network is divided in several management domains. In each domain there is a single mobile agent that remains stationary at one of the nodes, performing remote SNMP transactions with the other nodes of the domain. The aggregated results are then sent to the central node. Whichever the model, SNMP transactions with each managed node are sequential, in the sense that a new query to a specific node is issued only after receiving the response of the previous query to that node. However, in the SC and SD models, each manager conducts independent and parallel transactions with each managed node it deals with.

These are all familiar models: SC represents the majority of SNMP applications and SD is just a simple hierarchic delegation topology. Previous work [19-21] also proposes the MG, MD and MW models, although using different designations.

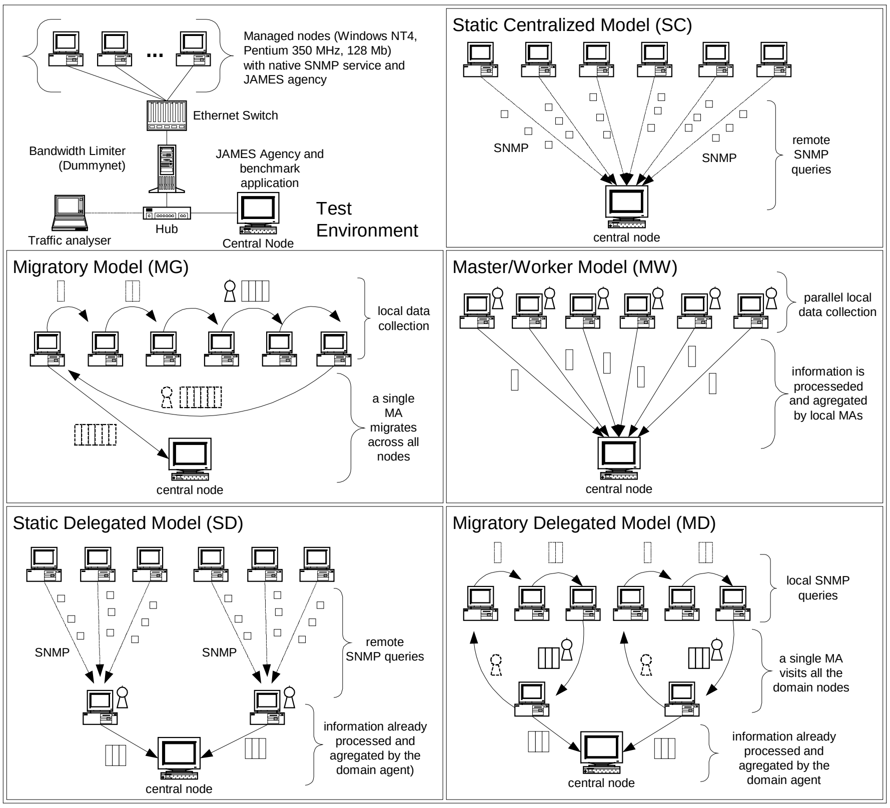


Figure 1: Test Environment and Distribution Models

The test bed consisted of relatively homogeneous Intel workstations (Windows NT 4, Pentium II 350 MHz and 128 Mbyte of RAM), connected to a switched *ethernet* LAN. The MA framework was provided by the JAMES platform [22]. Each managed node included the native Windows NT SNMP service and a JAMES agency, while the central node also included a JAMES agency.

Some tests were conducted using the switched 10 Mbps network, but more restrictive network conditions were also considered, placing a bandwidth limiter [23] between the central node and the managed nodes (Figure 1). Due to practical limitations the bandwidth between managed nodes was not restricted. Anyway, the SC and MW models have no communication at all between managed nodes and the delegated models (SD, MD) only have communication between nodes of the same delegation domain. Since delegation domains tend to be based on LAN boundaries, the test bed still represents the most common application environments.

Management data sent to the central node by mobile agents is previously compressed. For each five MIB-II objects retrieved from the managed node only the rounded arithmetic average is sent, along with the identifier of the first object. This simple method compresses data to 20% of its original size (only one “object” is sent, not the five retrieved objects) and was considered as representative of the computational effort and compression rate of more realistic techniques.

Several parameters were introduced during the benchmark tests, such as the number of managed nodes (from 1 to 120), the amount of involved management information (from 25 to 600 SNMP MIB-II INTEGER objects per managed node and per poll) and the network bandwidth between the central node and the managed nodes. Other parameters also introduced in the tests – such as optimization techniques to reduce the number of SNMP transactions or to further compress management data – will not be considered in this paper.

Measurements included both the performance of the system and the network traffic received and generated by the central node. The performance was measured by the time it takes since the central node triggers a new poll until the management data of every node is received and decoded at the central point. In order to isolate setup costs from “steady-state” running costs, the first poll and the subsequent polls (when stationary agents are already installed and migratory agents benefit from code caching mechanisms) were independently measured. Performance results presented in this paper represent the average of 20 “steady-state” polls, with a standard deviation within 5%. However, despite this 5% boundary, the nature of the test environment (tenths of NT workstations, native SNMP service and Java-based mobile agents) produced a small but noticeable fluctuation on the results, particularly for the fastest models.

3. Performance of Static and Constrained Mobility Models

3.1 The Effect of Locality

In order to measure the effect of locality in performance, a very simple test was conducted using two exactly equal machines (1 central node and 1 managed node) and adjusting the bandwidth between 32 Kbps and 4 Mbps. Experimental measurements show that the relative performance of the MW model, when compared to the SC model, improves with the progressive restriction of bandwidth and the increasing amount of management data (Figure 2, first chart).

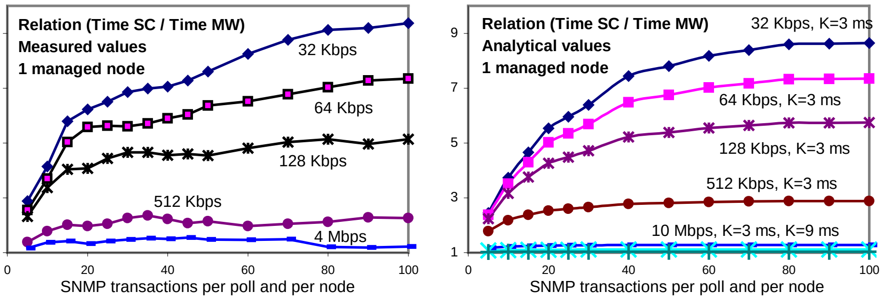


Figure 2: Performance Relation Between SC and MW (1 managed node)

However, this locality-associated speedup has a maximum limit determined by network latency, processing time and efficiency of data compression – as already mentioned, the implemented semantic compression reduces the amount of data to 20%, but there are also other factors that may reduce or increase network traffic. Expression 1 defines the expected response time of the SC model for a single managed node, and Expression 2 approximately defines the response time expected for the MW model. In these expressions *N* represents the number of SNMP transactions per poll and per managed node (in our case each transaction involves 5 MIB-II objects). $T_{getSNMP}$ and $T_{respSNMP}$ indicate the time needed to build an SNMP get-request PDU and the time needed to decode and process an SNMP get-response PDU, while T_{AgSNMP} represents the time it takes for the SNMP service to decode, process and reply to an SNMP request. Our experimental measurements pointed to a value around 3 ms for the sum of $T_{getSNMP}$, $T_{respSNMP}$ and T_{AgSNMP}.

$$T_{SC} = N \cdot \left(T_{getSNMP} + \frac{S_{get\_pdu}}{BW} + T_{AgSNMP} + \frac{S_{resp\_pdu}}{BW} + T_{respSNMP} \right) \quad (1)$$

$$T_{MW} = \frac{S_{setup}}{BW} + N \cdot \left(T_{getSNMP} + T_{AgSNMP} + T_{respSNMP} \right) + N \cdot \left(\frac{f_{compression} \cdot S_{resp\_pdu}}{BW} \right) \quad (2)$$

$$\frac{T_{SC}}{T_{MW}} = \frac{K \cdot BW + S_{get\_pdu} + S_{resp\_pdu}}{K \cdot BW + \frac{S_{setup}}{N} + f_{compression} \cdot S_{resp\_pdu}}, K = T_{getSNMP} + T_{AgSNMP} + T_{respSNMP}, \quad (3)$$

BW represents the available bandwidth; S_{get_pdu} and S_{resp_pdu} indicate the size of the SNMP packets (in this specific test 122 and 128 bytes, respectively); and S_{setup} is the size of the initial message sent by the central node to trigger the poll in the MW model (64 bytes). The relation between the size of the SNMP responses and the size of the data sent to the central node, in the MW model, is indicated by $f_{compression}$. Our traffic measurements showed that this factor changes according to the total amount of management data involved: it was 67% for 5 SNMP transactions, 44% for 10 transactions, 29% for 20 transactions, 21% for 40 transactions and 18% for 80 and 100 SNMP transactions. Although the semantic compression would point to a stable value of 20%, the communication overheads justify this variation. The second chart of Figure 2 represents the analytically expected relation between the performance of SC and MW models, according to Expression 3[*]. Despite some irregularities on the experimental measurements, there is a noticeable correspondence between experimental and analytical values.

The first relevant conclusion of these results is that for a given value of network latency there is a fixed limit on the relative performance speedup provided by sending a mobile agent to the managed node, and the only way to increase this limit is to further compress the management information. Although this is possible for some applications (e.g. autonomous monitoring with very small amounts of data sent to the central node) it may be difficult to achieve major improvements with other kinds of applications. Another conclusion is that the speedup provided by sending a MA to the managed node is not significant within LAN environments. With 512 Kbps this speedup reaches 289% but with 10 Mbps it stays below 10%.

3.2 The Effect of Distribution

When there is more than one managed node the central point potentially becomes the bottleneck. Even using asynchronous SNMP transactions (i.e. sending requests to nodes B and C while waiting for the answer of node A), as we did, the computational resources of the central node tend to get overloaded, leading to the degradation of response times. In this situation the MW model allows several mobile agents to conduct parallel SNMP transactions, therefore increasing the overall performance of the system. There still are some bottlenecks in the final transmission of the management data to the central point, but given the previous processing and compression of this data their effect becomes less relevant.

Figure 3 shows some of the measured response times for the SC and MW models (10 Mbps). The results show that the response time of the SC model is almost proportional to the number of managed nodes, indicating that the computational resources on the central point became a bottleneck. The MW model presents a much smaller degradation: the analysis of a more extensive set of measurements showed that, for the used test bed, the average performance degradation introduced by each additional managed node was just around 2,65% relatively to the performance obtained using a single managed node, i.e., the average response time managing 10 nodes is approximately 26% higher than the time obtained managing a single node. This degradation is related with the competition between each remote mobile agent in the transfer of the final results to the central point.

---

[*] Expression 3 uses the bandwidth to simplify the determination of network latency. In our experience there was a router between the managed node and the central node, and the bandwidth was controlled on both interfaces of the router. This duplicated the latency, specially affecting the SC model where sequential SNMP request/reply packets are sent over the network. For this reason, the values of Figure 2 consider that in the SC model the network latency is approximately *2S/BW*, rather than *S/BW*.

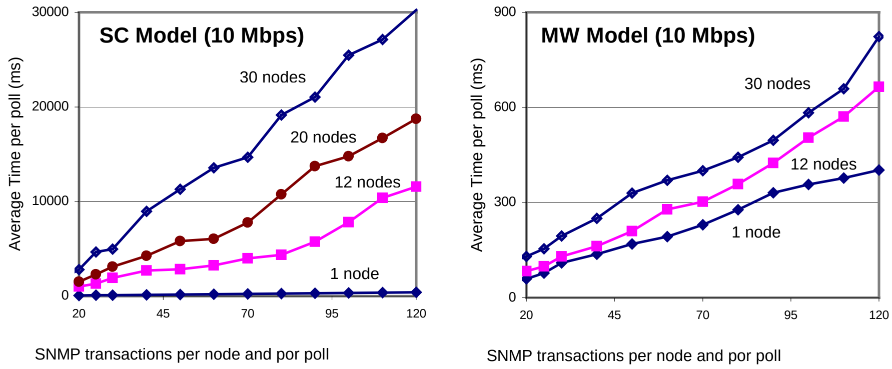


Figure 3: Performance of the SC and MW Models (several managed nodes)

These measurements are impressive but hardly surprising. They show that for local network environments it is distribution that boosts performance, not locality. This is an important conclusion because locality is an expensive feature: placing a mobile agent or some other kind of mobile code in each managed device is often unfeasible, costly or constrained by security and portability reasons. Distribution, on the other hand, can be provided at lower costs by using a strategically selected number of nodes, each managing a small part of the network (SD model). Furthermore, it is possible to adjust the level of distribution (number of delegated managers) according to the available resources and the desired performance.

Figure 4 shows some performance measurements for the SD model. The first chart shows how its performance can be adjusted – between the lower limit of SC and the upper limit of MW – by selecting an appropriated number of delegation domains. The second set of results shows that by using the SD model it is possible to increase the number of managed nodes with a very small degradation of performance, as long as the size of each delegation domain is kept constant. The average response time of the SD model with 120 nodes (12 domains with 10 nodes each) is less than twice the response time of the SC model with 10 managed nodes.

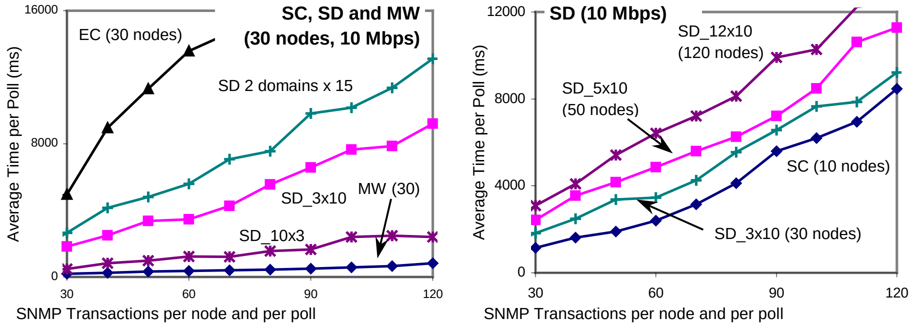


Figure 4: Performance of the SD Model

4. Performance of Mobility-based Models (MG, MD)

Section 3 discusses the effect of locality and distribution on the performance of management systems. In this Section a third factor is introduced: mobility. From the performance viewpoint, mobility is just an alternative way to achieve locality.

While the MW model is based on constrained mobility and strong distribution – one MA per managed node, where it remains stationary – migratory models rely on increased mobility – successive migrations taking place between managed nodes in order to locally perform each SNMP transaction – and less distribution (MD) or no distribution at all (MG). However, since agent migration is a relatively slow process, mobility-based models are expected to present additional performance costs, even considering migration speedup techniques.

The experimental measurements for those two models (Figure 5) confirm that in local network environments the performance of migratory models is not competitive, even when compared with the classic SC model. This is hardly surprising, given the already mentioned relatively small speedup associated to local SNMP transactions and the high costs of successive migrations. From a performance perspective, mobility-based models are not worthwhile unless the network becomes much slower (reinforcing the weight of locality) or extreme distribution is applied – by defining more and smaller delegation domains, as demonstrated by the response times achieved using 10 domains of 2 managed nodes each. However, even then, equivalent constrained-mobility models tend to perform better. Nevertheless, the MD model scales as well as the MW and SD models: by increasing the number of domains it is possible to increase the number of managed nodes without significant performance penalties. The second chart of Figure 5 shows how the average response times (using fixed size domains) remain in the same range while the network size increases six times.

5. Network Traffic

By performing local SNMP transactions, some models are able to process and aggregate management data before it is sent over the network, reducing its size directly at the source (e.g. MG, MD and MW). Mechanisms like semantic compression [19] significantly reduce the size of management information: compression rates around 5 to 1 are probably achievable in most practical situations, and even higher rates are not uncommon in monitoring applications.

In other models (e.g. SD), data compression occurs at an intermediate level: the domain manager. If the costs of intra-domain traffic are similar to the costs of inter-domain traffic this is not very interesting, since the global traffic will correspond to the addition of SNMP transactions (traffic levels in the same range of the SC model) and the final transmission of aggregated management data to the central point. However, if delegation domains cross no LAN boundaries, the costs of internal traffic probably become less relevant. For such environments, where communication costs are dictated almost exclusively by inter-domain links, the benefits of traffic compression provided by the SD model are similar to the benefits of locality-based models like MW.

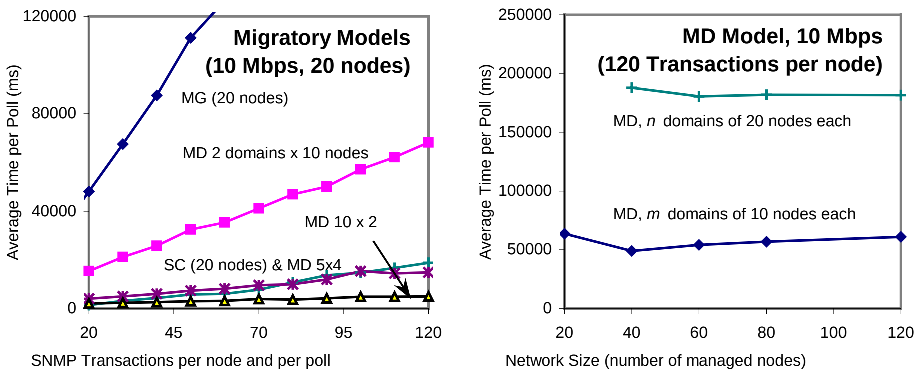
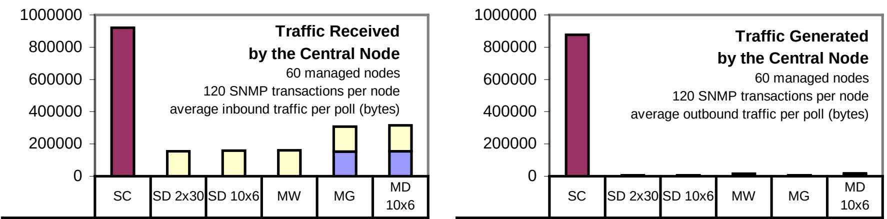


Figure 5: Performance and Scalability of Migratory Models (MG, MD)

Figure 6 shows the measurements of average incoming and outgoing traffic at the central node for one "steady-state" poll (i.e. setup costs of the first poll are not included), according to three categories: SNMP transactions; transfer of previously processed management data; and infrastructure control traffic generated by the supporting mobile agent platform. Intra-domain traffic is not included, but in the case of the SD model it is easy to extrapolate, since it equals the SNMP traffic of the SC model. The measurements highlight three different aspects: the effect of data compression; the weight of infrastructure control traffic in the migratory models; and the inefficiency of SNMP in the generation of outgoing traffic.

As already mentioned, our test application applies compression techniques that reduce management data to 20% of its original size. However, the effective traffic reduction in the transfer of management data – compared to raw SNMP transactions – is also influenced by communication overheads. Results presented in Figure 6 consider 120 SNMP transactions (600 SNMP objects) per node and per poll and, therefore, they have a very good compression rate: between 16,7% of the original size for the MG model – where all the data is sent by just one mobile agent – and 17,5% for the MW model – where there are 60 agents sending information. However, for smaller amounts of retrieved information, this rate gets worst. Table 1 shows how the effectively measured compression rates vary for different models.



	SC	SD 2x30	SD 10x6	MW	MG	MD 10x6
Mgm. Data	0	154256	158567	161027	154094	161110
SNMP	920811	0	0	0	0	0
Control	0	14	21	64	153437	154230

	SC	SD 2x30	SD 10x6	MW	MG	MD 10x6
Mgm. Data	0	4729	5767	15353	4608	7130
SNMP	877612	0	0	0	0	0
Control	0	522	619	1061	1342	9896

Figure 6: Network Traffic (measured at the central node)

SNMP transactions per poll	*5*	*10*	*20*	*40*	*80*	*120*
Compression rate (MW / SC)	67%	44%	29%	21%	18%	18%
Compression rate (SD 4x5 / SC)	33%	25%	22%	19%	18%	17%
Compression rate (SD 2x10 / SC)	24%	22%	19%	18%	17%	17%

Table 1: Effective Data Compression Rate (20 nodes, nominal rate: 20%)

The overhead of infrastructure control traffic for the migratory models is considerable, since the central node receives detailed notification whenever there is an agent migration. However, this traffic is related to the specific way the JAMES platform [22] was used in this test bed, and other implementations can easily reduce it using alternative or weaker infrastructure monitoring schemes.

The differences between SC and the other models in the volumes of generated traffic are related with the inefficiency of the SNMP protocol. Using SNMP it is necessary to send successive read requests (either *get* or *get-next* primitives) increasing outgoing traffic to unnecessary levels, at least when the more efficient *get-bulk* operation is unpractical. With distributed management it is possible to strongly reduce this outbound traffic or, at least, to keep it away from the central node – either performing local SNMP transactions or using the SD model. It should be mentioned, however, that presented traffic measurements correspond to the optimal situation where mobile agents already know in advance what management information they need to retrieve – the central manager just triggers the poll and acknowledges the reception of management data. Other applications may require more extensive flows of information from the central manager to the remote managers, reducing the efficiency gap between SC and other models.

6. Sensibility to Network Conditions

Performance measurements presented in Sections 3 and 4 were obtained in typical LAN environments (switched 10 Mbps) and, as already discussed, they essentially depend on the distribution of the management process. However, if bandwidth is gradually reduced, locality is expected to become, at some point, the most important influence on performance. To analyze this process, performance was also measured using slower links to the central point of the test bed.

In the migratory models (MG, MD) the bandwidth reduction showed no noticeable effect on performance. A possible explanation for this circumstance is that the degradation of the channel between managed nodes and the central point does not interfere with agent migration and, therefore, only the final transmission of management data sent to the central point is affected (control messages discussed in Section 5 are sent by the node the agent departs from and, consequently, impose no performance penalties). The relative system performance is already so bad that the degradation in the final data transmission is hardly relevant, at least in the test bed conditions (MG, MD 4x5 and MD 6x5 models; 20 or 30 managed nodes; 128 Kbps to 2 Mbps; 5 to 120 SNMP transactions per node and per poll).

The effect of available bandwidth is more perceivable for constrained mobility models. Figure 7 shows the average time per poll for the MW and SD models using

20 managed nodes. For relatively fast network links the performance remains nearly unaffected, since the additional delay in the final transmission of management data to the central point is relatively small. Considering the nominal bandwidth and the volumes of data transmitted, the expected degradation when moving from 2 Mbps to 1 Mbps, for instance, is around 180 ms, for 100 SNMP transactions (analytical values). This implies a degradation around 22% for the MW model and 6% for the SD model with 4 domains.

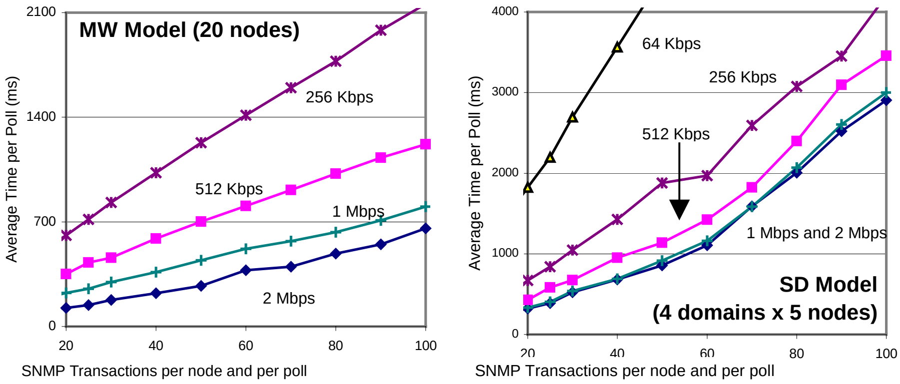


Figure 7: Effect of Available Bandwidth in the Constrained Mobility Models

However, the relative weight of this delay gradually increases to the point where it becomes the main factor in the global system performance. The measured performance of the MW model with 256 Kbps, for example, is approximately three times worse than with 2 Mbps. Performance starts to depend more on the network traffic and less on distribution. At some point, the processing time eventually becomes relatively irrelevant and the capacity of the communication channels becomes the limiting bottleneck. Figure 8 illustrates this situation by comparing the measured performance of the SC model with the maximum theoretical performance (based just on the available bandwidth and the network traffic, and assuming there where no processing delays). Similar results were achieved with the MW and SD models.

In this scenario of scarce bandwidth the performance of each model becomes as good as the data compression rates it provides. In this sense, locality assumes an important role because it makes possible to apply data compression at the source but "proximity" – in the way it is provided by the SD model – may also produce similar results, as long as the costs of intra-domain transmission are relatively smaller than the costs of transmission between the delegated manager and the central node. Anyway, the focus should be on the compression of management data before the most expensive network links: if data is compressed to half its size, with extremely slow links, it is eventually possible to reduce response times to values near half the original response times.

Another aspect that needs clarification is the general notion that the relative performance speedup of agent-based solutions – as well as DNM in general – gets better under unfavorable network conditions. These solutions are certainly very interesting in this kind of environments, since they potentially reduce network usage and, consequently, also improve the system performance, when compared to classical centralized solutions under the same conditions. However, one should keep in mind that their relative speedup under optimal network conditions may in fact be even better, because in this situation speedup is mainly determined by different factors, like distribution. Widely distributed applications that, for some reason, are unable to apply strong compression to management data will actually show worst relative performance – when compared to centralized solutions – as network conditions become more severe. This is a straightforward conclusion but, since most experimental measurements use small networks where distribution benefits are less relevant than data compression benefits, it is often overlooked.

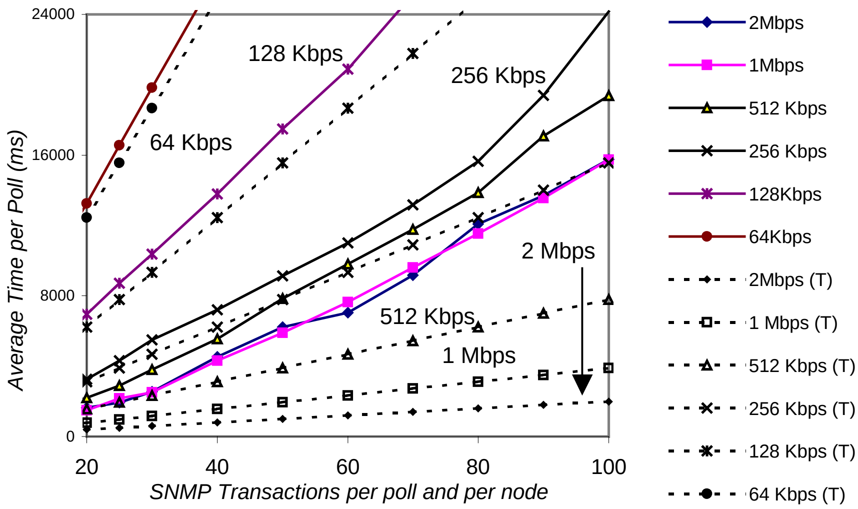


Figure 8: Measured and Theoretical Maximum Performance (SC, 20 nodes)

7. Setup Costs

Experimental measurements presented so far in this paper focus on the “steady-state” behavior of the system: stationary agents are already installed and code caches optimize the migration of mobile agents. It was implicitly considered that for the majority of practical applications the system would be stable enough to pay off the setup costs of each model. Nevertheless, it is important to know how long will it take to pay off those costs.

The effects of setup procedures differ from model to model. Migratory models (MG, MD) tend to be slower in the first poll, when code-caching mechanisms are still unable to enhance the migration process. However, the difference between the performance of the first poll and subsequent polls rarely exceeded 10% in our tests. For constrained mobility models, however, there is a much bigger difference between the first poll – when mobile agents are installed – and following polls.

Figure 9 presents the average accumulated response times for several models, considering 20 managed nodes and 30 and 120 SNMP transactions per node and per poll (respectively 50 and 600 SNMP objects). In the first case (50 SNMP objects) the absolute differences between each model are smaller, and therefore break-even for the fastest model (MW) is achieved just around the 25th poll. In the second case the higher amount of management data results in bigger differences between each model, and consequently break-even for the MW model is achieved earlier (around the 11th poll). However, it should be stressed that the centralized solution (SC) is hardly competitive, even considering a small number of polls.

A similar behavior was observed for network traffic. In our test bed, even considering 60 managed nodes and just 25 SNMP objects per poll and per managed node, setup costs are compensated around the 17th poll – for the less favorable MW model – or the 3rd poll (SD 15x4). For higher volumes of management data break-even occurs earlier (e.g. when retrieving 300 SNMP objects per node and per poll, compensation occurs right in the 1st poll).

Another aspect of setup costs is related to the requirements on the managed nodes. Decentralized models discussed in this paper rely on some kind of mobile agent support infrastructure distributed over the network. Compared with traditional SNMP services – usually a standard feature of managed devices – this mobile agent infrastructure needs more computational resources, is difficult to install and requires explicit maintenance. In some situations it is simply not possible, desirable or cost-effective to install and run such an infrastructure.

From this perspective the classic SC model is in advantage: it requires no mobile agent support at all. MG, MD and MW models, on the other hand, need mobile agent support from every managed node. The SD model is less demanding, since it requires mobile agent support only on the delegated domain managers. This is simpler to accomplish because the number of MA-enabled nodes becomes smaller (one per domain) and also for the reason that it is probably easier to find or add one capable node per domain than to extend agent support to every managed node – these nodes will often consist of embedded devices or proprietary equipment.

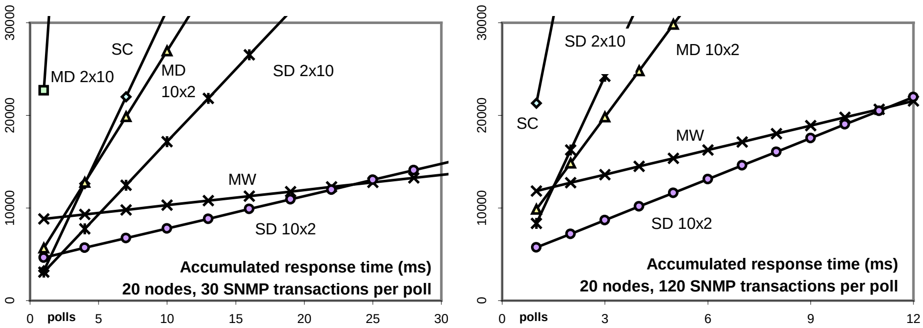


Figure 9: Accumulated Response Times (performance break-even points)

8. Conclusions

Over the last few years several studies addressed the efficiency of MA-based DNM systems. These studies have shown the potential advantages in fields like network traffic, performance, and scalability. However, despite this relative abundance of studies, we feel that the behavior of MA systems in typical network management scenarios is still not quite understood. This motivated us in the development of an experimental test bed that reproduces a large range of such "real-world" scenarios. Rather than focusing on a specific issue (e.g. migration costs or traffic compression mechanisms), in this paper we deliberately try to provide an overlook on a relatively large set of aspects and on the way they interact with each other. Obvious space restrictions prevented the detailed analysis and discussion of the obtained experimental results, but an overview of the behavior of MA-based management systems was provided – by presenting just a small but representative selection of available results. Probably, this overview does not bring new conclusions to previous work, but it does provide a different perspective on those conclusions.

One of the most important lessons we learned from the study was that for a large number of situations, including traditional LAN environments, performance gains are determined mainly by distribution rather than locality (local client/server transactions), which just provides marginal improvements. There is a general and correct perception that a mix of locality and distribution determines performance of mobile agent systems. However, we did not expect this mix to be so unbalanced in a plain simple switched 10 Mbps *ethernet*. This is very important because there are alternative ways of providing distribution, such as the SD model where performance speedup is adjustable according to the costs of deploying mobile agents across the network.

Naturally, with more severe network conditions the relative weight of distribution decreases, eventually reaching a point where performance is determined almost exclusively by the time it takes to send the management data over the network, opposed to the time it takes to process this data. In this scenario, locality plays an important role because it allows the compression of management data before network transmission. However, when intra-domain network connections are much better than connections between the delegation domains and the central manager, locality might be replaced by "proximity" (as provided by the SD model) without performance losses. In these situations, locality and proximity are just two different solutions to reduce the size of management data before it reaches the critical network links.

Traditionally, locality is also pointed out as a solution to overcome delays related with network latency (e.g. the successive request/reply sequences of SNMP when consulting distant nodes). However, if the manager simultaneously deals with several managed nodes, as usually happens, this latency becomes less relevant than network throughput.

Another lesson we learned was that, from a performance perspective, data compression is hardly relevant with good network conditions. In fact, some experiences (not presented in this paper) showed us that with 10 Mbps networks the usage of more aggressive compression techniques (e.g. providing an additional fourfold reduction on the size of management data with no increase on computational costs) provide very small performance improvements. However, once bandwidth becomes scarce the effect of data compression becomes much more noticeable, and each distribution model turns out to be as fast as the compression rate it provides. Previous work already explored interesting semantic data compression techniques [19-20], and in some favorable conditions the amount of data sent back to the central point is extremely reduced (e.g. autonomous fault detection and correction). However, even less elegant compression techniques, such as plain “zipping” of data, may provide good results.

Our last lesson was that migratory models show no performance gains or traffic reduction when compared with constrained-mobility systems. Although our test bed was based on a mobile agent infrastructure, only two of the studied models explicitly require mobile agents (MG, MD) and the other distributed models (MW, SD) use just constrained mobility. Naturally, this raises some doubts about the usefulness of MAT in typical application scenarios. In fact we believe that with the exception of niche applications – based on algorithms that do require extensive agent migrations – the use of MAT is not justifiable by speed or traffic optimization but rather by increased flexibility, scalability and robustness. Using sporadic MA migration and replication, it is possible to dynamically adjust the number and location of “static” delegated managers (mobile agents) according to network and host conditions. However, competing technologies also claim to provide this kind of functionality and, therefore, further qualitative and quantitative studies – such as [17] – will be necessary to evaluate the interest of MAT in this usage scenario. Nevertheless, the generic conclusions we present for MA-based constrained mobility models should also apply, without significant differences, to similar DNM systems built using alternative technologies.

References

[1] J.-F. Martin-Flatin, S. Znaty, “Two Taxonomies of Distributed Network and Systems Management Paradigms”, Technical Report DSC/2000/032, École Polytechnique Fédérale de Lausanne, Switzerland, 2000.

[2] Y. Yemini, G. Goldszmidt, S. Yemini, “Network Management by Delegation”, Proceedings of IFIP 2nd International Symposium on Integrated Network Management, Washington, 1991.

[3] M. Siegl, G. Trausmuth, “Hierarchical Network Management: A Concept and its Prototype in SNMPv2”, Computer Networks and ISDN Systems, Vol. 28, No. 4, pp. 441-452, 1996.

[4] J. Schönwälder, J. Quittek, C. Kappler, “Building Distributed Management Applications Using the IETF Script MIB”, IEEE Journal on Selected Areas in Communications, Vol. 18, Nº. 5, pp. 702-714, IEEE Comm. Society, May 2000.

[5] A. Schade, A. P. Trommler, “CORBA-based Model for Distributed Application Management”, Proceedings of the 7th IFIP/IEEE Workshop on Distributed Systems: Operations & Management (DSOM’96), 1996.

[6] A. Hayzelden, J. Bigham (Eds.), Software Agents for Future Communication Systems, Springer-Verlag, 1999.

[7] N. Anerousis, “Scalable Management Services Using Java and the World Wide Web”, Proceedings of the 9th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM’98), 1998.
[8] G. Aschemann, S. Domnitcheva, P. Hasselmeyer, R. Kehr, A. Zeidler, “A Framework for the Integration of Legacy Devices into a Jini Management Federation”, Proceedings of the Tenth IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM’99), Zurich, 1999.
[9] A. Bieszcad, B. Pagurek, T. White, “Mobile Agents for Network Management”, IEEE Communications Surveys, 4Q, 1998.
[10] M. Baldi, G. Picco, “Evaluating the Tradeoffs of Mobile Code Design Paradigms in Network Management Applications”, Proceedings of ICSE'98 – 20th International Conference on Software Engineering, 1998.
[11] M. Rubinstein, O. Duarte, “Evaluating Tradeoffs of Mobile Agents in Network Management”, Networking and Information Systems Journal, Vol. 2, No. 2, 1999, Hermes-Science Publications.
[12] A. Liotta, G. Knight, G. Pavlou, “On the Performance of Decentralised Monitoring using Mobile Agents”, Proceedings of the Tenth IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM’99), 1999.
[13] R. Pinheiro, A. Poylisher, H. Caldwell, “Mobile Agents for Aggregation of Network Management Data”, Proceedings of ASA/MA’99 – First International Symposium on Agent Systems and Applications and First International Symposium on Mobile Agents, pp. 130-141, Palm Springs, 1999.
[14] M. Zapf, K. Herrmann und K. Geihs, “Decentralized SNMP Management with Mobile Agents”, Proceedings of the IM’99, Boston, 1999.
[15] A. Puliafito, O. Tomarchio, “Using Mobile Agents to Implement Flexible Network Management Strategies”, Computer Communication Journal, 23(8), April 2000.
[16] A. Sahai, C. Morin, “Enabling a Mobile Network manager (MNM) Through Mobile Agents”, Proceedings of Mobile Agents’98 (MA’98), Stuttgart, 1998.
[17] C. Bohoris, A. Liotta, G. Pavlou, “Evaluation of Constrained Mobility for Programmability in Network Management”, Proceedings of the Eleventh IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM’2000), Springer-Verlag LNCS 1960, Austin, 2000.
[18] S. Lipperts, “How to Efficiently Deploy Mobile Agents for an Integrated Management”, Proceedings of the 3rd IFIP/GI International Conference on Trends towards a Universal Service Market (USM’2000), Munich, 2000.
[19] D. Gavalas, D. Greenwood, M. Ghanbari, M. O’Mahony, “An Infrastructure for Distributed and Dynamic Network Management based on Mobile Agent Technology”, Proc. of the IEEE International Conference on Communications (ICC’99), 1999.
[20] D. Gavalas, D. Greenwood, M. Ghanbari, M. O'Mahony, “Enabling Mobile Agent Technology for Intelligent Bulk Management Data Filtering”, Proceedings of the 2000 IEEE/IFIP Network Operations and Management Symposium (NOMS’2000), Honolulu, Hawaii, April 2000.
[21] D. Gavalas, D. Greenwood, M. Ghanbari, M. O'Mahony, “Implementing a Highly Scalable and Adaptive Agent-Based Management Framework”, Proceedings of the IEEE Global Communications Conference (Globecom’2000), San Francisco, 2000.
[22] P. Simões, L. Silva, F. Boavida, “Integrating SNMP into a Mobile Agent Infrastructure”, Proceedings of the Tenth IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM’99), 1999.
[23] L. Rizzo, “Dummynet: a simple approach to the evaluation of network protocols”, ACM Computer Communication Review, Vol. 27, No. 1, pp.31-41, January 1997.