

# On the Performance and Scalability of Decentralised Monitoring using Mobile Agents

Antonio Liotta<sup>1</sup>, Graham Knight<sup>1</sup>, George Pavlou<sup>2</sup>

<sup>1</sup>Department of Computer Science, University College London, Gower Street,  
London WC1E 6BT, United Kingdom  
{A.Liotta, G.Knight}@cs.ucl.ac.uk

<sup>2</sup>Centre for Communication Systems Research, University of Surrey, Guildford  
Surrey GU2 5XH, United Kingdom  
G.Pavlou@ee.surrey.ac.uk

**Abstract.** The problem of improving the *performance* and *scalability* of current monitoring systems, which generally follow a centralised and static management model is considered herein. Several alternative solutions based on Mobile Agents (MAs), specifically tailored to network monitoring are described. In particular, the key problem of computing efficiently the initial *number* and *location* of MAs is addressed, and techniques for rapid MA deployment are described. Mathematical models for the proposed MA solutions and for the common technique of centralised polling are built and a comparative analysis of their performance is carried out. Performance is expressed in terms of *monitoring traffic* and *delay*. Results on traffic show that MAs typically offer improved performance, although they do not always scale better than polling. However, results on delay demonstrate that MA solutions improve both scalability and performance under most circumstances. **Keywords:** Monitoring, Polling, Mobile Agents, Agents Cloning, Scalability.

## 1 Introduction

Current management systems follow a centralised and static management model although several researchers have reported the limitations of this approach [6-10]. Systems that follow this model often involve the transmission of large quantities of raw data from remote systems to a central management station before the data can be further processed, disseminated and presented. Consequently, these systems are subject to network congestion causing delays, drops in performance, and inability to react promptly to faults. Therefore, in view of the exponential growth in size, distribution, and complexity of communication networks, current management systems present severe limitations as far as performance, scalability, and flexibility are concerned.

In practice, network monitoring relies largely on *centralised polling*, which inherits most of these limitations. Polling is a process whereby an issuing node or *polling station* in a communication network broadcasts a query to every node and waits to receive a unique response from each of them [1]. Polling finds wide application in the control of distributed systems and in the management of distributed databases (see

references quoted in [1]). More generally, polling is used for a large class of applications that need to detect a variety of changing conditions and react to them, often subject to time constraints.

A potential approach to more decentralised, efficient, and flexible management solutions is to equip the management system with Mobile Agent (MA) capabilities [2, 7]. If management tasks are implemented using MAs, raw data can be pre-processed—e.g. filtered and aggregated—near the remote network devices. In fact, MAs can be located optimally in order to minimise network *traffic* (incurred by management) and monitoring *delay*. MAs can be organised in either a *hierarchical* or a *non-hierarchical* fashion. The former case tends to be particularly suitable for hierarchical networks—e.g. networks consisting of sub-networks. For example, monitoring tasks requiring first the *collection* and then the *aggregation* of raw data from each of the network levels—e.g. from each sub-network—can be implemented by distributing agents at each intermediate level. Each agent will be in charge of producing the required level of data aggregation for a specific sub-network and providing high-level information to the other agents in the hierarchy. Alternatively, a non-hierarchical MA organisation can be used, which is particularly suited to non-hierarchical networks and non-hierarchical monitoring tasks.

Another feature that can have a significant impact on the efficiency of monitoring systems is MA *cloning*, the ability of agents to create and dispatch copies, or ‘clones’, of themselves. Cloning can significantly reduce MA *deployment time* and *traffic*, which are critical factors in large-scale systems, and is particularly beneficial in the case of hierarchical networks. However, despite its benefits, cloning adds an extra degree of complexity to the MA system and is applicable only to a sub-class of monitoring tasks - e.g., tasks that can be decomposed into a finite number of identical sub-tasks.

Therefore, agent mobility and cloning can play a key role in the improvement of the efficiency of monitoring systems. In addition, the ability of agents to be re-programmed and re-configured dynamically introduces an attractive perspective of adding flexibility to the monitoring system. However, despite the great interest arisen in the application of MA technologies to Network and System Management [2-11], MAs have not yet found wide application for a number of reasons. For instance, MA systems are more complex than their centralised counterparts, particularly with respect to their design, implementation, and maintenance. Moreover, without reliable metrics by which to assess the benefits that such systems offer, it has been impossible to compare the overall value of different approaches.

Herein the problem of improving the performance and scalability of monitoring systems and quantifying the potential benefits introduced by MAs is addressed. Such evaluation requires the classification of potential MA solutions specifically tailored to network monitoring and the introduction of a suitable assessment methodology. Several alternative MA approaches are described in Sec.2. The method used to assess the performance and scalability of each technique and to compare them with centralised polling is described in Sec.3. Performance models for the specific solutions under study are presented in Sec.4 and, then, used in Sec.5 to carry out a comparative analysis of their performance and scalability. Modelling and analysis are restricted to the case of hierarchical networks.

The results of the present study are relevant to a wide range of applications since the problem of finding and assessing efficient, scalable, and flexible techniques to monitor distributed systems is not bound to the field of network monitoring.

Monitoring is a fundamental function used for performance, fault and security management and therefore it is of paramount importance for their efficiency. Similarly, system management, as well as generic distributed systems relying on communication networks, can benefit from MA technologies.

## 2 Mobile Agent Approaches for Network Monitoring

MAs can support each of the four main monitoring activities performed in a loosely-coupled, object based distributed system —*generation, processing, dissemination and presentation* of information [17] (pp.303-347). They can generate monitoring data in the form of *status* and *event reports*, according to different modalities. For example, status reporting can be either periodic or on request; events can be detected by MAs acting as *probes* and reported in a variety of different *formats*. MAs are able to monitor a set of MOs through polling, then process and deliver (directly or otherwise) the acquired data to a management station. Data processing techniques include operations such as *merging* of monitoring traces; *combination* of monitoring information, thus increasing the level of abstraction of data; *filtering* of monitoring data, reducing the volume of data; and *analysis* of monitoring information, producing data statistics.

In the following sub-sections we present a classification of potential MA approaches to network monitoring. Our classification is based on the adopted monitoring model, MA organisation, MA deployment pattern, and MA configuration algorithm. Specific MA solutions based on this classification are described in Sec.4 and analysed in Sec.5.

### 2.1 MA Monitoring Models (MA Types)

We assume that monitoring tasks are specified by giving a set of MOs and of monitoring parameters. When tasks are implemented with MAs the set of MOs is partitioned into disjoint subsets and each subset is monitored by a different MA.

We consider three possible monitoring models and their respective MA types. In the first model, based on *polling*, each MA periodically polls the subset of MOs associated to it and periodically analyses the results. Each MA is periodically polled by either other MAs or the monitoring station. The second model is based on *periodic notification*. Each MA periodically polls its MOs, analyses the results, and *notifies* either other MAs or the monitoring station. Finally, the third model is based on *event-driven alarm generation*. As in the previous cases, an MA periodically polls its MOs and analyses the results. However, MAs generate data (alarms) only if specific events are detected. Therefore, in each of these three models MAs first collect data through polling and then pre-processed this data. The models only differ in the inter-agent communication mechanism. In the remaining part of this paper MA Types associated to the first, the second, and the third monitoring model are referred to as MA Type A, B, and C, respectively.

## 2.2 MA Organisation

MAs can be organised in either a *hierarchical* or a *non-hierarchical* (or *flat*) fashion, depending on the way monitoring tasks are implemented. In hierarchical MA organisations, MOs are monitored by MAs belonging to the last hierarchical level. MAs at each level can communicate only with the MAs at one level above and one level below, according to one of the monitoring models specified in Sec.2.1. Thus, only MAs at the first level communicate with the monitoring station. On the contrary, in the flat MA organisation there is no inter-agent communication and each MA monitors its MOs, analyses data, and communicates directly with the monitoring station.

Flat MAs can be of type A, B and C. In hierarchical MAs we distinguish between the MAs belonging to the last level and the ones belonging to the other levels. The MA types admitted in this case are AA, AB, AC, BB, BC, CC, where the first letter refers to the type of MAs belonging to the last level and the second letter refers to the remaining MAs.

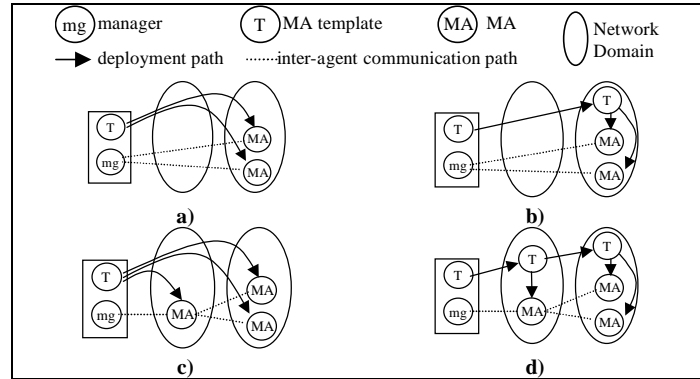
## 2.3 MA Deployment Patterns

Another factor that can be crucial to performance and scalability is the technique, or *pattern*, used to deploy MAs. The way MAs travel does not depend only on their characteristics and on their communication paradigms and protocols. MAs can also implement migration rules and strategies according to predefined migration patterns. Existing patterns, which can find effective application to network monitoring and can be used as MA deployment patterns, are described in [4].

However, within the scope of our analysis we consider only four different deployment patterns that are based on MA organisation and on MA *cloning* capabilities —where cloning is ability of agents to create and dispatch copies or ‘clones’ of themselves. In the case of MA incapable of cloning, MAs can only be created at the monitoring station.

The first deployment pattern is the ‘flat broadcast with no cloning’ pattern (Fig.1a). In this case MAs are organised in a non-hierarchical fashion. Each of them is created by the manager at the monitoring station by using one or more MA *templates*. A template is a model used to create an MA which implement a specific task or part of a task. Each MA is, then, associated to a subset of the MOs, and is dispatched to a target location. The target location can be either pre-defined by the MA’s creator or decided by the MA itself, depending on its degree of autonomy.

The second deployment pattern is the ‘flat broadcast with cloning’ one (Fig.1b). In this case, the manager dispatches a single MA (the MA template) which, in turn, clones the actual MAs. Each MA is then configured - e.g., its target MOs and monitoring parameters are set - and dispatched to an appropriate location. The third pattern is the ‘hierarchical broadcast with no cloning’ one (Fig.1c). All MAs are created by the manager at the monitoring station, like in the ‘flat broadcast with no cloning’ pattern. However, MAs are organised in a hierarchical fashion. Finally, the ‘hierarchical broadcast with cloning’ pattern is similar to the ‘flat broadcast with cloning’, but differs from it in the way MAs are organised.



**Fig. 1.** MA deployment patterns: **a)** ‘flat broadcast with no cloning’ capabilities; **b)** ‘flat broadcast with cloning’ capabilities; **c)** ‘hierarchical broadcast with no cloning’; **d)** ‘hierarchical broadcast with cloning’ capabilities.

The patterns admitting cloning (Fig.1b and Fig.1d) are generally preferable since they can result in a significant reduction of MA deployment traffic and time. This is a consequence of the fact that in most situations a single MA template is transmitted between intermediate MA locations instead of a set of MAs. An example showing a typical application of cloning is a case in which the network links involved in the MA deployment process are not uniform in bandwidth and latency. In fact, in this case the transmission of several MAs through congested links might be impractical. However, cloning cannot be used in all circumstances: it is applicable only to a sub-class of monitoring tasks - e.g., tasks that can be decomposed into a finite number of identical sub-tasks. Moreover, cloning requires a relatively complicated management of MAs, which in turn results in additional overheads.

The patterns involving hierarchical deployment (Fig.1c and Fig.1d) are particularly useful not only in the case of hierarchical networks, but also more generally when the monitoring task can be organised in a hierarchical fashion. On the other hand, flat patterns (Fig.1a and Fig.1b) represent a simpler alternative for non-hierarchical networks or monitoring tasks.

## 2.4 MA Configuration

The configuration of MAs involves two main problems: 1) the determination of the *number* of MAs suitable to a given monitoring task; 2) the determination of the *location* of each MA.

The MA configuration problem is analogous to the problem of locating multiple emergency facilities in a transport network and, more specifically, is equivalent to trying to minimise the total distance between these facilities and the nodes in the network. This problem has been studied extensively and is usually referred to as the *p*-median problem or the *minisum location problem* (see [14-16] and references quoted therein).

Unfortunately, the *p*-median problem on a general network is NP-hard [15]. The approaches to solving it can be classified into five categories [14]: 1) enumeration, 2) graph theoretic, 3) heuristics, 4) primal-based mathematical programming, and 5)

dual-based mathematical programming. The enumeration of all possible solutions to determine the optimal one is unfeasible for large networks. Graph-theoretic approaches take advantage of the network structure to determine the p-median and are very efficient for tree networks for which algorithms of polynomial complexity have been developed. For example Goldman has presented an algorithm that solves the 1-median problem in  $O(N)$  steps [13], where  $N$  is the number of nodes in the network. For the p-median problem an  $O(N^2p^2)$  algorithm is suggested in [16] and an  $O(N^2p^2)$  one is proposed in [15].

On the other hand, heuristics procedures cannot guarantee an optimal solution but can be applied to any general network structure ([14] pp.55-58). Very little computational work has been reported on this approach. Nevertheless, the complexity of heuristics procedures is usually smaller than that of the other approaches. Finally, the mathematical programming approaches have attracted wide attention and have proved rather successful for general networks ([14] pp.58-68).

Despite the similarities between the p-median problem and the MA configuration problem, they have different algorithmic constraints and priorities. For instance, the former problem aims at finding solutions as close to optimality as possible, trying to minimise the computational time. Instead, in the latter case it is of paramount importance to minimise the traffic incurred into the network by the configuration algorithm as well as to solve the problem in a relatively shorter time.

In the remaining part of this section we sketch the description of two possible solutions, for the case of MAs capable of cloning and incapable of cloning, respectively. Both solutions make use of the routing information obtainable from network routers and are based on the assumption that such information is available to managers and MAs. We also assume that MOs belong to an Intranet and that MA hosts - i.e., locations in which MAs are entitled to run - are evenly distributed in the network. This, in other terms, means that for each router there is always an MA host that is located relatively close to it and, for each LAN, the number of MA hosts is proportional to the number of MOs belonging to the LAN. Under these assumptions, the MA distribution tree —i.e., the set of routes used for MA deployment— does not differ significantly from the routing tree rooted in the monitoring station. Without loss of generality, we envisage a scenario in which routers can act as MA hosts during MA deployment. In such case, the MA distribution tree would actually coincide with the routing tree.

**Configuration of MAs Capable of Cloning.** Initially a monitoring task is given by defining a set of monitoring parameters, such as a set of MOs, the required monitoring accuracy, a set of functions applied to process the monitored data, and so on. The monitored parameters are then analysed in order to determine the next-hop or hops that need to be followed in order to reach the objects. If  $n_h$  is the number of alternative routes,  $MO$  is partitioned into  $n_h$  disjoint sets,  $MO_1, MO_2, \dots, MO_{n_h}$ ,  $n_h$  MAs are cloned and each one is associated to a different  $MO_i$  and deployed through  $next\_hop_i$ . Then, each MA decides whether to keep cloning other MAs or start monitoring its subset of  $MO$ . This decision is based on the analysis of the local routing tables. In particular, an MA is cloned for each next-hop, which is also an MA host. Then, this process continues recursively until the last hierarchy of MA host has been reached and, thus, MA deployment has been completed. Therefore, in this case the number of MAs,  $m$  is the result of this recursive mechanism; it is not pre-

determined at start up time. Note that there are several possible variations to this algorithm. We have described a deployment scheme for hierarchical MA organisations. If they reflect a flat organisation no MAs remain in a switching device if a next-hop MA host is found. Thus, MAs are cloned only if more than one next-hop is found. Otherwise the MA simply migrates to the next-hop. Other variations can reflect different conditions for the interruption of the recursive cloning mechanism. For example, an MA may base its decision on whether or not to keep cloning upon additional conditions such as estimates on the number of MOs in its monitoring domain.

**Configuration of MAs Incapable of Cloning.** The number of MAs,  $m$  is determined at start up time and is based on information which we assume is available to the manager, such as the number of MO domains - e.g., internet domains - and the number of MOs per domain.  $m$  MAs are generated at the monitoring station and a subset of the given MOs is associated to each of them, according to a procedure which is analogous to the one adopted for cloning-enabled MAs. Finally, each MA follows the distribution tree according to its target MOs and to the routing information extracted at each router. Each MA stops migrating towards its target MOs when it fails to find a unique next-hop destination. Then it starts monitoring its target MOs.

### 3 Method of Evaluation

The methodology used to assess monitoring systems is based on a mathematical, comparative analysis of the performance and scalability of various MA solutions against the common technique of centralised polling. Performance models of typical MA solutions based on the classification described in Sec.2 are included in Sec.4. In the following subsections we describe the basic assumptions on the network model, and the metrics and method used to assess performance and scalability.

**Network Model.** We have adopted the *network model* described in [1]. The network is modelled by a connected graph  $G=(V,E)$  with the vertices corresponding to nodes (processors) and the edges corresponding to communication links, which are modelled by the *All Ports-Full Duplex* communication model. This network model has been widely used because it generally reflects the hardware characteristics of networks (see references quoted in [1]).

**Metrics.** The *traffic* incurred by the monitoring system and the monitoring *delay* are modelled using an approach similar to the one described in [18]. *Traffic* is modelled as the sum of packet hops incurred by monitoring —i.e., the number of edges traversed by monitoring packets— multiplied by their respective packet size,  $b$  and packet rate,  $P_r$ . The traffic  $T(v_1, v_2)$  between any two points  $v_1, v_2 \in V$  subject to a bit rate  $B_r$  is  $T(v_1, v_2) = B_r * d(v_1, v_2)$ , where  $B_r = b * P_r$ ,  $b$  is the size of a poll request or response, and  $d(v_1, v_2)$  is the distance between  $v_1$  and  $v_2$ . The *distance* in the network is measured using the "hop" metric, in which each edge has unit weight. Finally, the *delay*,  $D(v_1, v_2)$  between any two points,  $v_1$  and  $v_2$  is measured as the distance  $d(v_1, v_2)$ .

The *monitoring delay* is expressed as  $\max\{D(v_1, v_2)\}$  - i.e. the maximum number of time units required to perform a complete 'request-response' operation - where  $v_1, v_2 \in S \subseteq V$  and  $S$  includes the set of monitored nodes.

The *performance* of monitoring systems is expressed in terms of the above traffic and delay functions. In particular, the *relative monitoring traffic* and the *relative monitoring delay* are calculated for each of the proposed MA solutions. Thus, for a given monitoring solution, both the former and the latter are expressed as a percentage of their 'naive-polling' counterparts.

*Scalability* is defined in [12] as the ability to increase the size of the problem domain with a small or negligible increase in the solution's time and space complexity. For the purposes of our investigation, scalability is specifically defined as the ability to increase the number of monitored entities  $N$ , the polling rate  $P_r$  or the network diameter  $D(u)$  - i.e., the maximum distance between any two nodes in the network - with a small or negligible decrease in performance. The *scale* of a given monitoring problem is measured in terms of  $N$ ,  $P_r$ , and  $D(u)$ .

**Analytical Evaluation.** The analysis includes the definition of typical MA solutions and the evaluation of the lower bounds on traffic and delay for each of them. Lower bounds on traffic and delay are also calculated for two solutions based on centralised polling: the naive approach used in practical system, and the optimal one described in [1]. *Optimal polling* differs from *naive polling* in the broadcasting of poll requests, which in the former case is very similar to the 'hierarchical broadcast with cloning' pattern depicted in Fig.1d.

These lower bounds on traffic and delay are first used to calculate the above defined relative traffic and delay and, then, to assess scalability. We consider that the factors limiting the scale of a monitoring system are the traffic incurred by it and the delay in gathering data. Consequently, scalability is assessed by studying the order of the above traffic and delay functions for  $N$ ,  $P_r$  or  $D(u)$  tending to infinity.

## 4 Performance Models for Network Monitoring

In this section we describe mathematical models for two different solutions based on centralised polling and for several typical solutions based on MAs. These models are used in Sec.5 to carry out a comparative analysis of their performance and scalability. Since the main focus of this paper is to identify potential MA solutions rather than providing a detailed description of the mathematical models, most mathematical details have been omitted. The interested reader may refer to [19] for further details. Moreover, in order to simplify the complexity of the mathematical analysis, the models provided below assume that the network admits at least one *n-ary* routing tree rooted at the monitoring station,  $u$ . This simplifying assumption is in accordance with the properties of the routing trees of typical network topologies reported in [1]. In particular, the following models assume routing trees having  $d(u)$  binary sub-trees, where  $d(u)$  is the order of the node which hosts the monitoring station.



#### 4.1 Naïve Centralised Polling

A polling operation can be modelled as a two-step process. First, a polling station  $u$  broadcasts a query to every node in the network that is being monitored. Then, each node sends a unique response back to  $u$  [1]. Thus, we can express traffic,  $T$  as the sum of two terms, the broadcast traffic  $T_b$  and the gathering traffic  $T_g$ ,

$$T = T_b + T_g = 2 * \sum_{j=0}^{D(u)-1} \sum_{i=j}^{D(u)-1} d(u) * P_r * b * 2^i \quad (1)$$

Where  $d(u)$  is the *degree* (or valence) of  $u$  in  $G$  —i.e. the number of edge ends at  $u$ —  $P_r$  is the *Polling Rate* —i.e. the number of polling operations per unit of time—  $b$  is the size in bits of a poll request and response (assumed equal for simplicity), and  $D(u)$  is the network diameter. After some calculations we obtain the following expression:

$$T = 2 * d(u) * P_r * b * (1 + (D(u)-1) * 2^{D(u)}) \quad (2)$$

This approach is the most widely used in current systems and is termed ‘*naive centralised polling*’ within the scope of this paper.

The delay can be expressed again as the sum of two terms, the broadcast time and the gathering time. The lower bound on delay for general network topologies, as reported in [1] is:

$$D = P_r * \left\{ D(u) + \max \left\{ \left\lceil \frac{N-1}{d(u)} \right\rceil, D(u) \right\} \right\} \quad (3)$$

For the binary case  $N=d(u)*(2^{D(u)}-1)$  and, a lower bound on delay is

$$D = P_r * \left\{ D(u) + 2^{D(u)} - 1 \right\} \quad (4)$$

However, in this case the exact delay can be expressed as

$$D = 2 * P_r * \left\{ 2^{D(u)+1} - D(u) - 2 \right\} \quad (5)$$

#### 4.2 Optimal Centralised Polling

A more efficient polling algorithm has been presented and proved optimal in [1]. In this case the polling station  $u$  sends a request only to its neighbours  $N(u)$  which, in turn, duplicate the request and forward it to their respective neighbours. The process continues until all nodes have been reached by a request. Therefore, the broadcast traffic will be smaller than its naive-polling counterpart, and the resulting incurred traffic can be expressed as

$$T = d(u) * P_r * b * (D(u) * 2^{D(u)}) \quad (6)$$

The lower bound on delay for general network topologies is reported in [1]. For the binary case and assuming that  $D(u)$  is relatively large, delays can be expressed as:

$$D = P_r * \left\{ 2^{D(u)} + 1 \right\} \quad (7)$$

### 4.3 Mobile Agent Solutions: Traffic Models

The total traffic,  $T$  incurred in the network by MAs is modelled as the sum of three terms, the *deployment traffic*, the *collection traffic*, and the *delivery traffic*. The first term accounts for the traffic incurred during MA deployment. The second term represents the traffic incurred when MAs gather information from MOs. The third term includes the traffic involved in the communication among MAs and between MAs and the monitoring station. Each of these terms may vary, depending on the MA type, organisation, deployment pattern, and configuration algorithm. In particular, the MAs' ability to reduce traffic is modelled by the MA *selectivity*,  $\sigma$ , defined as the ratio between the amount of data collected and delivered by an MA respectively. The selectivity is larger than one for MAs implementing tasks which perform data filtering or aggregation and is typically significantly larger than one. Lower bounds on the traffic incurred by three different MA solutions are expressed below. Finally, expressions for the selectivity of Type A, B, and C MAs are given. Further mathematical details of the following models are reported in [19].

#### Flat MA, Type A, B or C, Incapable of Cloning.

$$T = \frac{B * d(u)}{2 * O_p} * L * 2^L + P_r * b * d(u) * \left[ \left( \frac{D(u) - L - 1}{4} + \frac{L}{4 * \sigma} \right) * 2^{D(u)} + 2^L \right] \quad (8)$$

Where  $B$  is the size in bits of MAs,  $O_p$  is the duration of the monitoring task,  $L$  is the network hierarchical level of MAs, and the remaining variables are as specified above.

#### Hierarchical MA, Type A, B or C, Incapable of Cloning.

$$T = \frac{B * d(u)}{O_p} * \left\{ (L-1) * 2^L + 1 \right\} + P_r * b * d(u) * \left[ \left( \frac{D(u) - L - 1}{4} + \frac{1}{4 * \sigma_p} \right) * 2^{D(u)} + \frac{\sigma_q + 1}{\sigma_q} * 2^L + \frac{2}{\sigma_q} \right] \quad (9)$$

Where  $\sigma_q$  and  $\sigma_p$  are the selectivity of MAs at the intermediate and at the last hierarchical level, respectively.

#### Hierarchical MA, Type A, B or C, Capable of Cloning.

$$T = \frac{B * d(u)}{O_p} * (2^L - 1) + P_r * b * d(u) * \left[ \left( \frac{D(u) - L - 1}{4} + \frac{1}{4 * \sigma_p} \right) * 2^{D(u)} + \frac{\sigma_q + 1}{\sigma_q} * 2^L + \frac{2}{\sigma_q} \right] \quad (10)$$

**MA Selectivity.** The delivery traffic can be dramatically affected by the way MAs perform monitoring operations. The choice of MA monitoring model significantly affects their *selectivity*  $\sigma$ . If we distinguish between Flat configurations ( $p$ MAs,  $\sigma=\sigma_{p,x}$ ) and hierarchical configurations ( $q$ MAs,  $\sigma=\sigma_{p,x}$ ), the selectivity for *Type A*, *B*, and *C* MAs, in the case of binary sub-rooting trees, can be expressed as follows:

$$\begin{aligned}\sigma_{p,A} &= \frac{2 * P_r * b * (|N_x| + 1)}{2 * P_r * b} = 2^{D(u)-L-1}; \quad \sigma_{q,A} = \frac{2 * P_r * b * 2}{2 * P_r * b * 1} = 2; \\ \sigma_{p,B} &= \frac{2 * P_r * b * (|N_x| + 1)}{N_r * b} = \frac{2 * P_r}{N_r} * 2^{D(u)-L-1}; \quad \sigma_{q,B} = \frac{N_r * b * 2}{N_r * b} = 2; \\ \sigma_{p,C} &= \frac{2 * P_r * b * (|N_x| + 1)}{\wp_p * O_p * b} = \frac{2}{\wp_p * O_p} * P_r * 2^{D(u)-L-1}; \quad \sigma_{q,C} = \frac{\wp_p * O_p * b}{\wp_q * O_p * b} = \frac{\wp_p}{\wp_q}\end{aligned}\tag{11}$$

where  $|N_x|$  is the maximum number of nodes monitored by one MA,  $\wp_p$  and  $\wp_q$  represent the probability of an alarm being raised within any unit of time by  $p$ MAs and  $q$ MAs respectively, and  $O_p$  is the duration, in units of time, of the monitoring task. Note that  $\sigma_{p,B} = (2 * P_r / N_r) * \sigma_{p,A}$ ,  $\sigma_{q,B} = \sigma_{q,A}$  and, since  $N_r < P_r$ ,  $\sigma_{p,B} > 2 * \sigma_{p,A}$ . In addition,  $\sigma_{p,C} = [2 / (\wp_p * O_p)] * P_r * \sigma_{p,A}$  and  $\sigma_{p,C}$  is a very large constant if we assume that the system does not generate relatively frequent alarms ( $\wp_q \ll \wp_p$ ). Thus, assuming that  $P_r \propto N_r$ , the order of both  $\sigma_{p,A}$  and  $\sigma_{p,B}$  is  $O(2^{D(u)}, N)$  whereas  $\sigma_{p,C}$  is  $O(P_r * 2^{D(u)}, P_r * N)$ . Therefore, we can expect a comparable traffic behaviour for *Type A* and *B* MAs and a better traffic behaviour for *Type C* MAs.

#### 4.4 Mobile Agent Solutions: Delay Models

The total monitoring delay,  $D$  is modelled as the sum of three terms, the MA *deployment delay*, the data *collection delay*, and the data *delivery delay*. Lower bounds on the monitoring delay for five different MA solutions are expressed below. We assume that the transmission of ‘polls’ and ‘responses’ takes one unit of time, the transmissions of MAs takes 10 units of time, and an MA cloning operation takes 10 units of time, for each traversed link. The number of MAs is  $m=d(u)*2^{L-1}$  in the case of Flat MA solutions and  $m=d(u)*(2^L-1)$  in the case of hierarchical solutions, where  $L \in \{1 \dots D(u)\}$ .

##### Flat MA, Type A, Incapable of Cloning.

$$D = 10 * (2^L - 1) + P_r * (2^{D(u)-L} - 2) + P_r * 2^{L-1}\tag{12}$$

##### Flat MA, Type A, Capable of Cloning.

$$D = 20 * L + P_r * (2^{D(u)-L} - 2) + P_r * 2^{L-1}\tag{13}$$

##### Hierarchical MA, Type A-A, Incapable of Cloning.

$$D = 10 * (2^{L+1} - L - 2) + P_r * (2^{D(u)-L} - 2) + P_r * L\tag{14}$$

### Hierarchical MA, Type A-A, Capable of Cloning.

$$D = 20 * L + P_r * (2^{D(u)-L} - 2) + P_r * L \quad (15)$$

### Hierarchical MA, Type C-C, Capable of Cloning.

$$D = 20 * L + P_r * (2^{D(u)-L} - 2) + \wp * L \quad (16)$$

where  $\wp$  is the probability that MAs generate alarms.

## 5 Comparative Analysis

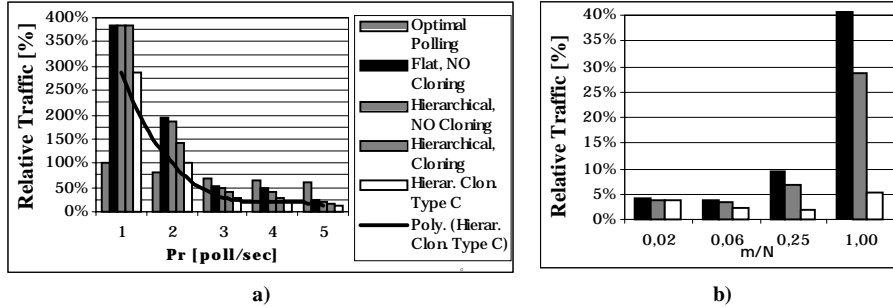
In this section we discuss the performance and scalability of the monitoring solutions modelled in Sec.4: MA solutions, centralised naïve polling, and centralised optimal polling. The analysis is carried out in two parts: the former is based on traffic functions; the latter is based on delay functions.

### 5.1 Analysis Based on Traffic

**Performance.** The *relative traffic*,  $R_r$ , incurred by a given monitoring solution can be expressed as the percentage of the traffic that would be incurred by its 'naive polling' counterpart solution. That is,  $R_r = (T/T_{poll}) * 100$ , where  $T_{poll}$  is the 'naive polling' traffic and  $T$  is the traffic incurred by the solution under comparison. The relative traffic of the 'optimal polling' solution and of four different solutions based on MAs is depicted in Fig.2. An important result is that for each of the considered solutions  $R_r$  diminishes very rapidly with scale and settles at 10-20% or even lower, depending on the MA pattern and configuration.

The behaviour of  $R_r$ , determined using the *polling rate*,  $P_r$ , as scale indicator is shown in Fig.2a. Analogous trends are obtained adopting the *network diameter*,  $D(u)$ , or the *number of monitored entities*,  $N$ , as scale indicators. The theoretical minimal value of  $R_r$  can be calculated by substituting the traffic expressions from equations 2, 6, 8, 9, and 10 into the expression of  $R_r$ . Notice that for all the MA solutions under study this value depends on a function of  $\text{Log}(m/N)$  and is independent of the selectivity. This shows the importance of MA configuration in terms of number and location of agents. For example, a relatively large number of MAs may be required to reduce data collection traffic; however, MA deployment traffic may lead to significant performance degradation when  $m \approx N$  (Fig.2b).

The effects of cloning and of the monitoring model can be observed from Fig.2. Cloning can lead to a significant reduction in MA deployment traffic, whereas the event-based model (*Type C* MAs) tends to reduce delivery traffic. Another interesting result is that MAs do not always represent the best solution. For instance, the first two sets of data in Fig. 2a show a specific monitoring problem for which none of the proposed MA solutions perform better than centralised polling. More generally, the analysis of equations 2, 6, 8, 9 and 10 highlights the conditions under which polling tends to be preferable, typically for relatively small values of  $P_r$  and  $O_p$ .

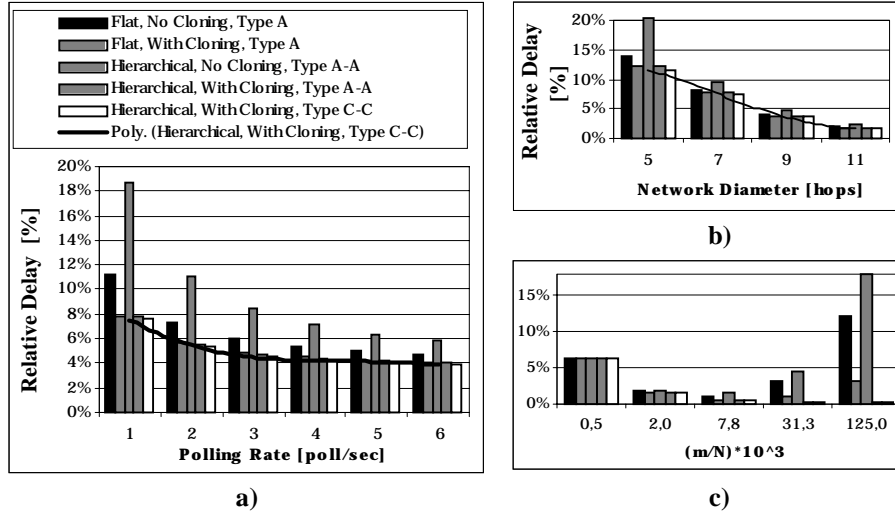


**Fig. 2.** Traffic incurred by different solutions, expressed as a percentage of the traffic generated by a 'Naive Centralised Polling' solution.  $D(u)=15$ ;  $P_r=10$ ;  $d(u)=3$ ;  $N=49149$ . All MA solutions are of Type A unless otherwise indicated. **a)** Traffic versus Polling Rate; **b)** Traffic versus the ratio between the number of MAs ( $m$ ) and the number of monitored nodes ( $N$ ).

**Scalability.** The analysis of equations 2 and 6 reveals that over increasing scales of the monitoring problem —i.e., for increasing values of  $N$ ,  $P_r$ , and  $D(u)$ — both naïve polling and optimal polling traffic functions operate at  $O[N, P_r, D(u)]$ . If the expressions of  $\sigma_{p,x}$  and  $\sigma_{q,x}$  of equation 11 are substituted into equations 8, 9, and 10, we can conclude that type A, B, and C MAs operate at  $O[N, P_r, D(u)]$  too, regardless of the difference in their selectivity. This is due to the fact that selectivity only affects delivery traffic, whereas the predominant term is collection traffic. Therefore, MAs do not generally scale better than polling, as shown by the behaviour of the polynomial approximation of a 'hierarchical with cloning Type C-C' solution, which does not tend to zero for  $P_r$  tending to infinity. However, under some specific circumstances MA hierarchical solutions can scale better. For instance, when  $[L=D(u)-1]$  traffic functions operate at most at  $O[N, P_r]$  for type A and B MAs and at  $O[N]$  for type C MAs.

## 5.2 Analysis Based on Delay

**Performance.** The *relative monitoring delay*,  $R_D$ , of a given monitoring solution can be expressed as a percentage of the delay of its 'naive polling' counterpart solution. That is,  $R_D=(D/D_{poll})*100$ , where  $D_{poll}$  is the 'naive polling' monitoring delay and  $D$  is the delay of the solution under comparison. The relative delay of the 'optimal polling' solution and of five different solutions based on MAs is depicted in Fig.3. A notably important result is that, for the chosen MA configuration parameters, the relative delay is always smaller than the 'naive polling' delay for all the proposed MA solutions, and decreases very rapidly with scale.



**Fig. 3.** Monitoring Delay, expressed as a percentage of the delay obtained with a ‘Naïve Centralised Polling’ solution. All MA solutions are of Type A unless otherwise specified. **a)** Relative delay versus polling rate. The relative delay obtained with a ‘centralised optimal polling’ solution (not shown) is 25.6%.  $D(u)=8$ ;  $d(u)=3$ ;  $N=765$ ;  $L=\text{FLOOR}(D(u)/2.5, 1)$ ;  $m=12$  for flat solutions;  $m=21$  for hierarchical solutions. Probability of event generation, for type C MAs is  $\text{FLOOR}(P/4, 1)$ . **b)** Relative delay versus network diameter. The relative delay obtained with a ‘centralised optimal polling’ solution (not shown) is in the range [25-29%].  $P_r=5$ ;  $d(u)=3$ ;  $N=[45-6141]$ ;  $L=\text{FLOOR}(D(u)/2.5, 1)$ ;  $m=[3-24]$  for flat solutions;  $m=[3-45]$  for hierarchical solutions. Probability of event generation, for type C MAs is  $\text{FLOOR}(P/4, 1)=1$ . **c)** Relative delay against the ratio between the number of MAs ( $m$ ) and the number of monitored nodes ( $N$ ). The relative delay obtained with a ‘centralised optimal polling’ solution (not shown) is 25%.  $D(u)=12$ ;  $P_r=7$ ;  $d(u)=3$ ;  $N=12285$ ;  $L=\text{FLOOR}(D(u)/2.5, 1)$ ;  $m=[3-1536]$  for flat solutions;  $m=[3-3069]$  for hierarchical solutions. Probability of event generation, for type C MAs is  $\text{FLOOR}(P/4, 1)=1$ .

Cloning can lead to a significant reduction in MA deployment time, as well as deployment traffic, at the expenses of a more complicated deployment technique. The benefits of cloning are particularly evident for relatively small values of  $P_r$  and  $O_p$ , that is when the monitoring system is more sensitive to the MA delivery overheads (traffic and delay). Examples of this effect are shown in Fig.3a and Fig.3b, in which significant differences among MA solutions are obtained only for small values of  $P_r$ . Hierarchical solutions are particularly sensitive to cloning since they typically involve deploying a relatively large number of MAs.

The impact of the number of MAs on performance can be observed in Fig.3c. Solutions having a relatively large number of MAs tend to be characterised by high performance gains. However, increasing the number of MAs results in large deployment times, which often require the use of cloning.

Finally, the ‘hierarchical with cloning solution’ (type C-C) has been shown to represent the most efficient solution in general (Fig.3). However, ‘flat with cloning’ solutions have comparable efficiency in several cases and present the advantage of smaller deployment overheads.

**Scalability.** The analysis of equations 5 and 7 reveals that, over increasing scales of the monitoring problem, both 'naïve polling' and 'optimal polling' delay functions operate at  $O[P_r, 2^{D(u)}]$ . On the other hand, the analysis of equations 11-15 reveals that MA solutions scale at most as  $O[P_r, 2^{[(k-1)/k]D(u)}]$  for  $L < D(u)$  and as  $O[P_r, 2^{L-1}]$  for  $L \sim D(u)$ , where  $k=L/D(u) \geq 1$ ; hierarchical MA solutions scale as  $O[P_r, D(u)]$  for  $L \sim D(u)$ . Two main conclusions can be drawn from this analysis. Firstly, if  $P_r$  is used as scale indicator MA solutions do not represent a more scalable solution than 'naive polling', although they result into smaller monitoring delays in typical cases. Secondly, if either  $N$  or  $D(u)$  are used as scale indicators, MA solution always scale better than their 'naive polling' counterpart solution.

Similar conclusions can be drawn from Fig.3a and Fig.3b, which show the behaviour of  $R_D$  versus  $P_r$  and  $D(u)$  respectively. The behaviour of  $R_D$  versus  $N$  (not shown) is analogous to the one in Fig.3b. In Fig.3a the polynomial approximation of a 'hierarchical with cloning Type C-C' solution tends to a positive constant, in accordance with the finding that in this case there is no improvement in scalability. In contrast Fig.3b the polynomial approximation tends to zero, showing the improvement in scalability.

## 7 Conclusions

In this paper we have addressed the problem of improving the performance and scalability of network monitoring systems and quantifying the benefits obtainable with Mobile Agents against the more common technique of centralised polling. Results on traffic show that MAs typically offer improved performance, although they do not always scale better than polling. Nevertheless, in very-small scale monitoring problems, particularly for relatively small values of  $P_r$ , and for relatively short monitoring tasks, centralised polling tends to be preferable. Moreover, results on delay demonstrate that MA solutions improve both scalability and performance in most circumstances. Therefore, we believe that the results of this analysis provide quantitative evidence to support the application of MA technologies to network monitoring, provided that other issues related to their feasibility are addressed.

The modelling and analysis reported in this paper are restricted to the case of hierarchical networks such as telecommunications networks. This choice is motivated by two main reasons. Firstly, hierarchical networks are very common. Secondly, this case can be treated analytically, in contrast with arbitrary network topologies. It is conceivable that MAs may be beneficial also in the case of general networks such as the Internet. For this reason our future work is to extend the investigation to this case through simulations. This approach will allow the refinement of the network model, the introduction of new metrics for the assessment of performance, and the study of further aspects of code mobility.

## Acknowledgements

We are grateful to Hewlett-Packard for their sponsorship. We thank Keith Harrison and other researchers at HP Laboratories Bristol (HPLB), for providing interesting and stimulating ideas.

## References

1. Rescigno, A.: Optimal Polling in Communication Networks. In IEEE Transactions on Parallel and Distributed Systems, Vol.8, N.5 (1997)
2. Baldi, M., Picco, G.P.: Evaluating the Tradeoffs of Mobile Code Design Paradigms in Network Management Applications. In Proc. of ICSE'98, Kyoto, Japan, April 19-25 (1998).
3. Chiariglione, L.: Foundations for Intelligent Physical Agents. FIPA 98 Draft Specification, part 11, August 17 (1998). (<http://drogo.cselt.it/fipa/spec/fipa98/fipa98.html>)
4. Aridor, Y., Lange, D.B.: Agent Design Patterns: Elements of Agent Applications Design. In Proc. of Second International Conference on Autonomous Agents (Agents '98), May (1998)
5. Liotta, A., Knight, G., Pavlou, G.: Modelling Network and System Monitoring Over the Internet Using Mobile Agents. In Proc. of IEEE/IFIP NOMS'98, Vol.2, Feb.(1998) 300-312
6. Kahani, M., Beadle, H.W.P.: Decentralised Approaches for Network Management. Computer Communications Review, ACM SIGCOMM, Vol. 27 N.3 July (1997)
7. Baldi, M., Gai, S., Picco, G.P.: Exploiting Code mobility in Decentralised and Flexible Network Management. In Proc. of Workshop on Mobile Agents, Berlin, April (1997)
8. Goldszmidt, G.: Distributed Management by Delegation. PhD Thesis, Columbia University, NY, (1996)
9. Yemini, Y., Goldszmidt, G. G., Yemini, S.: Network Management by Delegation. In Proc. of Integrated Network Management II, Amsterdam (1991)
10. Goldszmidt, G., Yemini, Y.: Delegated Agents for Network Management. In IEEE Communications Magazine, Vol.36 No.3, March (1998).
11. Liotta, A., Knight, G.: Decomposition Patterns for MobileCode-based Management. In Proc. of HP-OVUA 1998. ENST de Bretagne, Rennes, France, April 19-21 (1998)
12. Casavant, T.L., Singhal, M., Readings in Distributed Computing Systems. In IEEE Computer Society Press, ISBN 0-8186-3032-9 (1994)
13. Goldman, A.J.: Optimal Center Location in Simple Networks. In Transportation Science, Vol.5 (1971) 212-221.
14. Handler, G.Y., Mirchandani, P.B.: Location on Networks Theory and Algorithms. The MIT Press (1979)
15. Kariv, O., Hakimi, S.L.: An Algorithmic Approach to Network Location Problems - Part 2: The P-medians. SIAM J. Appl. Math., Vol.37 (1979) 539-560
16. Matula, D.W., Kolde, R.: Efficient Multi-Median Location in Acyclic Networks. ORSA/TIMS Bulletin, No.2 (1976)
17. Sloman, M.: Network and Distributed Systems Management. Addison-Wesley (1994)
18. Zegura, E.W., Calvert, K.L., Donahoo, M.J.: A Quantitative Comparison of Graph-based Models for Internet Topology. IEEE/ACM Transactions on Networking. In press.
19. Liotta, A., Knight, G.: Efficient Network Monitoring Using Mobile Agents. Research Notes RN/99/6, Computer Science Department, University College London, January (1999).