Managing Mobile Agents with SNMP

Rui Pedro Lopes[(1)], José Luís Oliveira[(2)]

[(1)]Instituto Politécnico de Bragança, ESTiG, 5300 Bragança, Portugal

[(2)]Universidade de Aveiro, DET, 3810 Aveiro, Portugal

rlopes@ipb.pt, jlo@det.ua.pt

Keywords:

Network management, Mobile agents, SNMP

Abstract

This paper describe a project that integrates SNMP into a mobile agent environment in order to achieve a simple but powerful goal: mobile agents have to manage and being managed through SNMP.

1 Introduction

For several years the network management buzzword was mostly associated with SNMP. Guided by the simplicity and the shorter inference principles soon has conquer the attention of a market with a big appetite for this solutions. However, its evolution has suffered from several drawbacks and has open space for other approaches. The straight path that was maintained by SNMPv3 working group may have provide a new breath into the SNMP management framework. Beyond its recent enrichments, namely the security model, the enormous base of legacy knowledge and legacy systems leads the SNMP management framework to a necessary choice in management scenarios.

Although, current management systems seems not to accomplish well the deep effects of the Internet evolution which is expected to introduce into the net enormous quantities of new different equipment and services. New paradigms must be available to deal with the pressures of these new demands. Mobile agent technology has been exploited as one of such keys.

A Mobile agent is an autonomous program that can move across networks, from node to node, and assume agent behaviour, i.e., act on behalf of users or other entities [1]. Most of the research efforts on this topic have been developed under the context of the telecommunications market. Examples of such research have been: management distribution and delegation [2], network services deliver [3] network traffic optimisation and network's fault tolerance [4].

Some management operations require large bandwidth from the network, due to the necessity to transmit, for instance, tabular information. The SNMP is not well designed to deal efficiently with this type of transfer load. By changing the paradigm "move data" to "move code" mobile agents seems to save precious network resources. However, to be useful for network management, mobile agents must also be managed. An anarchist mobile agent can do worse than no agent at all.

This paper investigates and promotes the inclusion of SNMP in a mobile agents environment, as a method to empower agents with SNMP capabilities or as the way to remotely access and control agents.

2 Network management concepts

Network management issues were dominated over the last decade by two main approaches: the OSI Management Framework or shortly the CMIP/GDMO model and the IETF Management Framework (typically identified as SNMP). Due to its simplicity

SNMP soon gained a larger set of followers relegating the CMIP to a second and also insignificant market share (due to its powerful capabilities supporters still remain, specially across telecommunication operators).

The purpose of a network management system is to provide a basis for remote administration of network components and hosts. The basic structure of a standard network management system contains several managed nodes, which provides remote access to management instrumentation. These nodes are traditionally called managed agents. In addition, it contains at least one entity with management applications, typically called Network Management Station (NMS), which concentrates all the data processing functions. The information exchanged between agents and the NMS are conveyed by a management protocol.

The SNMP framework defines both the structure and attributes used to face with managed resources (SMI and MIB), and the way these resources are retrieved (SNMP) [5]. It does not define upper level tasks such as interpretation, correlation, and corrective measures. This type of decision must be performed by specific management applications [6].

3 Mobile Agents

As computer systems evolve new features are added turning the system more and more complex. Far goes the time when computers were exclusive of teams of experts. Nowadays, users with diverse abilities and different levels of knowledge may use computer and network systems. As result, the number of tasks and the number of issues to keep track is continuously growing. In this context, users usually feel a need for extra eyes, hands, time and even brainpower.

In this scenario agent technology can make the difference. With the ability to perform actions on behalf of other programs or on behalf of the user, agents can be used on areas such as information retrieval [7], electronic commerce [8], personal mobility, telecommunications and user interfaces [9] just to name a few. A particular kind of an agent with the ability to move is called a mobile agent.

Mobile network agents are pieces of code that can be dispatched from one computer and transported to a remote computer for execution. Arriving at the remote computer, they present their credentials and obtain access to local services and data (maintaining the previous execution state). The remote computer may also serve as a broker by bringing together agents with similar interests and compatible goals, thus providing a meeting place at which agents can interact.

Nowadays there are several kinds of technological solutions for the development of mobile agent based systems. All of them provide a platform to support the migration and execution of code (agent platform) and use scripting or interpreted languages (such as TCL or Java) to survive over heterogeneity [4].

Independently of the technology, a mobile agent is composed of two pieces: the code, which describes the agent behaviour, and the execution state. Comparing an agent with some real life example, we can think of an agent as a pizza delivery boy. The instructions are to deliver pizzas and the state is the number of pizzas left to distribute. When a mobile agent travels, it carries both the code and the state with it, so it not only knows what to do but it also remembers what it has done.

Mobile agents have a well define lifecycle composed of the states and transitions that it can perform (Figure 1).

According to the lifecycle, the agent management comprises the following items: creation, removal, suspension, resumption, cloning, copying, migration, saving, reloading and action invocation.

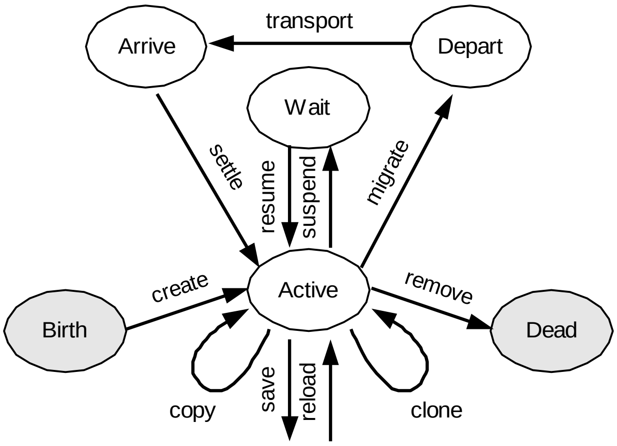

Figure 1 – Mobile Agent lifecycle.

3.1 MASIF

To ensure interoperability between mobile agent platforms, the OMG (Object Management Group) promoted a specification, MASIF [10], that we hope will gain a significant adherence. MASIF is based on CORBA standards and define methods to be implemented by different mobile agent platforms.

Behind the MASIF specification there are four fundamental mobile agent concepts (Figure 2):

?? **Agent –** The agent is the central entity in MASIF standard, the one for which the hierarchy was made to. Typically it has mobile characteristic, although it can also be a static agent. Beyond its mission, programming each agent maintains, at least, a set of attributes such as the agent authority, name, location and state.

?? **Place –** The Place concept provides the layer two of this model. Places are execution environments, which defines contexts in each Agent System (such as access control).

The Place is associated with a location, which consists of a place name and the address of the agent system. When a client requests the location of an agent, it receives the address of the place where the agent is executing. The Place-Agent association is a 1:M relation.

?? **Agent System –** The agent system (AS) is a platform to create, interpret, execute, transfer and terminate agents. It is the first physical entity in this model and it is identified by attributes such as name and address. Inside an AS can coexist several Places (a 1:M relation). This concept is many times known as Agency.

?? **Region –** A set of AS that shares the same authority. A region fully interconnects agent systems within its boundaries and enables the transfer of information between them. Each region contains one or more region access points i.e., regions may be interconnected to form a network. Also a 1:M relation is foreseen between a Region and AS.

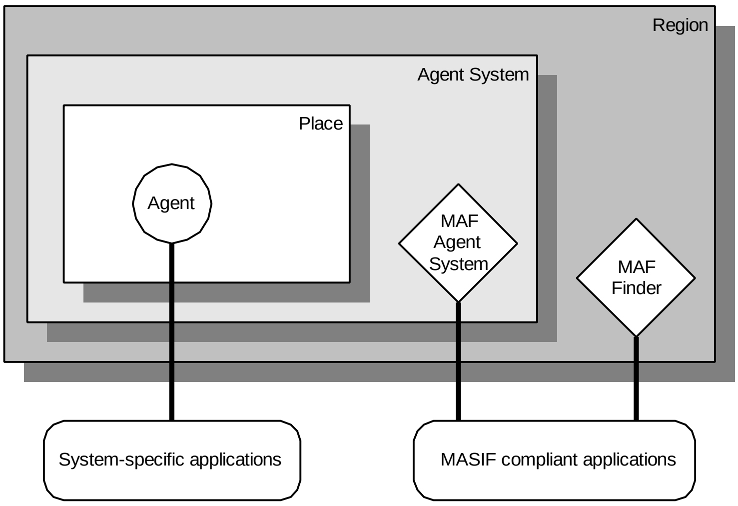

Figure 2 – MASIF architecture.

Although MASIF interfaces allows locating, creating, suspending or resuming agents and agent systems there are some missing aspects of the agent lifecycle. For example, it is not possible to force an agent to move, or to create a place inside some specified agent system. These operations must be performed by system-specific services.

3.2 Mobile Agents for Network Management

Several approaches consider mobile agents for managing networks and services due to the distributed nature, efficiency savings, traffic reduction and robustness [11][12][13]. Besides adding flexibility and a possibility to improve management efficiency this strategy also brings some difficulties, namely it increases the agent management difficulty [14] and also introduces some kind of usability challenges and possibly threats [15]. This leads to the fact that mobile agents also need to be managed.

By being MASIF compliant it is ensured that, at least, the mobile agent platform is as interoperable as the standards allow. It comprises several important aspects, such as agent management, transport, naming and tracking.

These two bases, the SNMP framework on one hand and the mobile agent platform on the other, provide the main foundation for the following approach.

4 SNMP Mobile Agent

Before addressing this section main issue it is important to establish a clear distinction between several concepts that use the same acronym (MA): Mobile Agents, Managed Agents and Management Agents.

A Mobile Agent is a program or process that uses some set of functions or methods (API) that allows it to transfer the code and the current state to other location. Mobile agents are usually executed in a specific environment (mobile agent platform), which also provides security features, cloning and copying capabilities. At this time, we choose to consider that a mobile agent is composed only by mobility aspect and leave the agency aspect for latter on. For now on we identify it as a *MobAg*.

SNMP Agents are computer entities responsible for collecting and storing management information local to the node and responding to requests for this information from the management station via a management protocol. For these reasons, they implement some set of MIB modules and an SNMP engine (according to naming conventions from [5]). The agent is as simple as possible so that it can be embedded in restricted environments, such as bridges or hubs. For now on we call it a MN (managed node).

A Management Agent is a program or process (agent) that provides an abstraction to network resources with the purpose of facilitating management operations. It usually acts on behalf of another program or application, performs instrumentation operations and executes management operations. Note that a management agent can implement some SNMP module and thus have characteristics of SNMP agent. From this point on we will use the term *ManAg* to name it.

4.1 Mobile SNMP Agent Architecture

There are two different aspects in the architecture. The MN aspect deals with the “act as an SNMP agent” characteristic and the *MobAg* aspect introduces mobile capabilities (Figure 3). From a top down approach, there are two ways to reach a Mobile SNMP Agent from the network: 1) from the SNMP engine and 2) from the Mobile Agent platform [16].

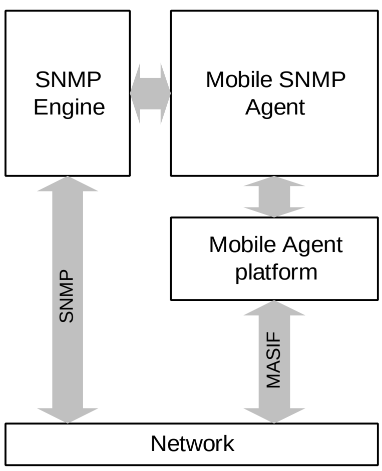

Figure 3 – Mobile SNMP Agent architecture.

In 1) the Mobile SNMP Agent must deal with MIB modules, distinguish Object Identifiers (OIDs) and process SNMP PUDs. The number and type of MIB modules are implementation dependent, so it is necessary some mechanism that allows the simple integration of more modules. Object Oriented languages allow using inheritance as a way to "hide" complexity in upper classes, so this is a nice starting way. Before addressing MIB instrumentation, it is necessary to parse the SNMP PDU fields (Figure 4).

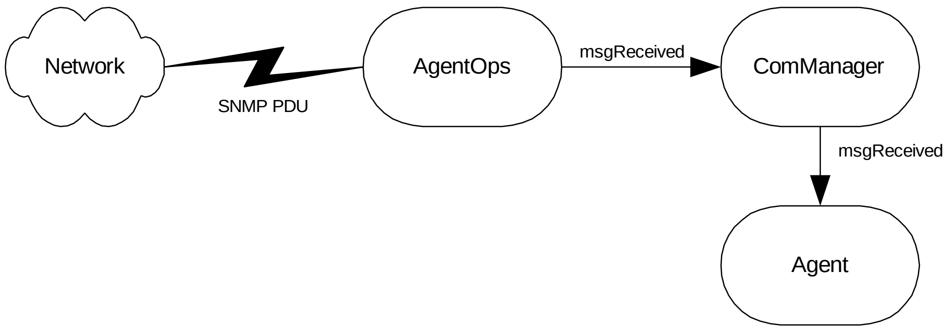

Figure 4 – PDU processing unit.

An independent thread is constantly monitoring the network for incoming PDU – `AgentOps` object. As a PDU arrives, `AgentOps` notifies the `ComManager`, which, in turn gives the PDU for the `Agent` to process.

`Agent` is the class that connects the PDU processing unit to the instrumentation, so it has some particularities. For each received PDU it checks the type of operation (*get*, *set*, *get-next*, *get-bulk*) and builds a list of SNMP variable bindings (OID plus value). For each and every variable binding it calls a method, `get(…)` or `set(…)`, according to the SNMP command. This is where the communication module connects to the instrumentation procedures.

Considering the SNMP operations and the tree-like organization of objects in the agent, some decisions can be made to help on the agent architecture planning.

Management operations have information about “which” object and “what” to do with it. In “which”, it is possible to point precisely the object (the case of *get* and *set*) and to define a walking procedure (*get-next* and *get-bulk*). In “what”, the operations are retrieval (*get*, *get-next* and *get-bulk*) and restore (*set*).

Adapting these concepts to an O-O language, the “which” is modeled by a container class (`Agent`) and the “what” are methods to call on contained objects (`Object`) (Figure 5).

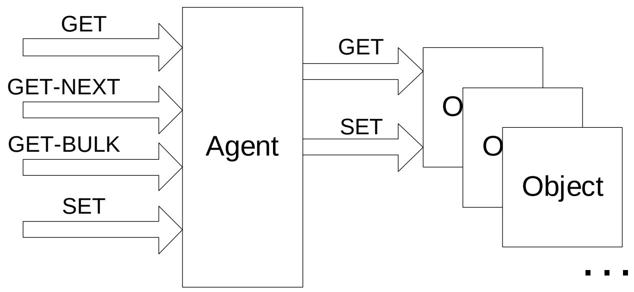

Figure 5 – SNMP operations on Agents.

The way that the `Agent` chooses which method and which module to call is based on two mechanisms: 1) OID object containment and 2) object specialization by inheritance. In 1), an `Agent` base class overrides a method that returns a reference to a specific MIB module object. The method simply inspects what is the name of the respective MIB module and checks the member variables for the appropriate object. When found, it returns a reference to the object.

4.2 Mobile Agents and SNMP – Integration scenarios

The coexistence of mobile agent technology with SNMP equipment and management tools must be grounded upon some rules: the SNMP is, for now, the framework of choice from a large number of equipment (hardware, software and technical knowledge), so any near upcoming management architecture must deal with SNMP standards and not the opposite.

Considering this coexistence, several scenarios can be foreseen:

Mobile Managers

On this scenario mobile agents should have the ability to perform management operations over local or over remote MIB:

a) The obvious solution seems to be the inclusion of a complete manager engine inside the *MobAg*, though the considerable increase on complexity and size of the code to move. This looks close to a traditional SNMP manager that was enhanced with mobile capabilities.

b) The SNMP services can be provided by the agent system and its usage shared amongst several *MobAg*.

c) If the access is restricted to local requests a special interface can be designed in order to provide a more efficient solution [17].

Mobile Managed Agents

The management of mobility is another issue that, at a first glance, appear to not imply integration between SNMP and mobile agents, i.e., the management of mobile environment can be provided by itself. However, there are several advantages on the SNMP side, since there are several network management systems that can the used also to control SNMP compliant mobile agents.

a) As for managers the solution to empower management operations over *MobAg* can be the inclusion of an SNMP agent engine in the *MobAg*. Again, the overcharge of the agent main functionality can be disastrous.

b) The SNMP agent services (get, set, trap, …) can be provided by the agent system. The AS provides a single entry point for the Mobile Agent MIB that is composed by the individual's contributions of the *MobAg*'s management information.

4.3 MIB Objects

In an environment where multiple mobile agents coexist there is a need to promote a uniform management situation. The *MobAg* may be autonomous different from each other but there are some aspects common to all of them. Although someone works in a chair factory and someone else is a clock maker both have to eat and have to sleep.

The AS provides the SNMP engine to the *MobAg* thus freeing the agent from unnecessary overcharge. The *MogAg* simply plugs in the information as a MIB module/extension.

This approach has some difficulties: 1) how do the manager knows that some AS has a particular information plugged in? 2) how to plug in the information?

In 1) the manager will know if it programmed the *MobAg* with the actions to perform, so it will be expected to behave in a known way.

In 2) the AS can provide an access interface to the included SNMP agent. This agent can support the AgentX protocol and thus allowing the extension of the local MIB with the plug in MIB. A simpler approach for 2) is the AS to provide SNMP services accessible from the *MobAg* via method invocation.

According to the previous discussion of mobile agent platforms management, a MASIF-compliant MIB was defined to reflect *MobAg* states and to allow simple management operations. It includes three different groups. One of them is related to platform objects management such as agents, AS and places (Figure 6).

The table `mamAgentSystemTable` registers all the AS registered on the region. The `mamPlaceTable` allows not only listing the places but also creating and removing places. The other object on this group (`mamAgentTable`) allows listing and managing agents currently registered in the same region. It may be useful for a management station to know specific agent information, such as the agent name, description, the name of the host agent system and the place where it is running. Other non-specific information may also be convenient, such as the existing agent systems and respective places.

Figure 6 – Mobile agent platform MIB.

Another group in this MIB module is related to agent history information. It can produce some notifications for operations performed by the *MobAg*.

The `mamTraps` group includes a table to define the recipients of SNMP traps. At the moment, for the sake of simplicity it does not allow to associate traps to specific operations.

5 Conclusions

Network management has been led by IETF SNMP framework, during the past years. SNMPv3 revised specifications have been recently approved as by the IESG and despite some limitations that can be pointed yet to this framework, there is now a great expectation upon it, after the SNMPv2 failure.

Mobile agents seem to be adequate to ever-changing environments, increasing system flexibility and robustness.

This paper presents an ongoing work over an SNMP framework based on mobility. The system allows the mobile agents to adapt in a changing environment, increasing management efficiency, reducing management traffic and providing better use of network resources while maintaining compatibility with existing standards. With this approach mobile agents platforms can be remotely managed through any commercial network management systems.

Furthermore, we expect to bring to light the real benefits and shortcomings of mobile technology on this particular environment.

6 References

[1] V. Pham, A. Karmouch, "Mobile Software Agents: An Overview", IEEE Communications, Vol. 36, No. 7, July 1998, pp. 26-37.

[2] G. Goldszmidt, Y. Yemini, "Delegated Agents for Network Management", *IEEE Communications Magazine*, Vol. 36 No. 3, pgs. 66-71, March 1998.

[3] S. Krause, T. Magedanz, "Mobile Service Agents enabling Intelligence on Demand in Telecommunications", *Proc. IEEE GLOBCOM'96*, 1996.

[4] R. P. Lopes, J. L. Oliveira, "Software Agents in Network Management", *Proc. of the 1st International Conference on Enterprise Information Systems – ICEIS'99*, March 1999, Setubal, Portugal.

[5] J. Case, R. Mundy, D. Partain, B. Stewart, RFC2570 (I), "Introduction to Version 3 of the Internet-standard Network Management Framework", April 1999.

[6] B. Wijnen, D. Harrington, R. Preshun, RFC2571 (DS), "An Architecture for Describing SNMP Management Frameworks", April 1999.

[7] I. Nekrestyanov, T. O'Meara, A. Patel, E. Romanova, "Building Topic-Specific Collections with Intelligent Agents", *Proc. of the 6th International Conference on Intelligence in Services and Networks*, IS&N'99, Barcelona, Spain, April 1999.

[8] M. Gleizes, A. Léger, E. Athanassiou, P. Glize, "Self-Organization and Learning in MultiAgent Based Brokerage Services", *Proc. of the 6th International Conference on Intelligence in Services and Networks*, IS&N'99, Barcelona, Spain, April 1999.

[9] S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers, R. Evans, *Software Agents: A review*, May 1997.

[10] MASIF, Mobile Agent System Interoperability Facilities, http://ftp.omg.org/pub/docs/orbos/97-10-05.pdf

[11] Perpetuum Mobile Procura, http://www.sce.carleton.ca/ netmanage/perpetum.shtml

[12] Collective Intelligence, http://www-iasc.enst-bretagne.fr/PROJECTS/SWARM/

[13] A. Bieszczad, B. Pagurek, T. White, "Mobile Agents for Network Management", Carleton University, Canada, 1997.

[14] M. Breugst, S. Choy, "Management of Mobile Agent Based Services", *Proc. of the 6th International Conference on Intelligence in Services and Networks*, IS&N'99, Barcelona, Spain, April 1999.

[15] E. Kaasinen, "Usability Challenges in Agent Based Services", *Proc. of the 6th International Conference on Intelligence in Services and Networks*, IS&N'99, Barcelona,

[16] J. Oliveira, R. Lopes, "Distributed Management based on Mobile Agents", *Proc. of the 1st International Workshop on Mobile Agents for Telecommunications Applications* – MATA'99, October 1999, Ottawa, Canada.

[17] B. Pagurek, Y. Wang, and T. White, "Integration of Mobile agents with SNMP: Why and How", ftp://ftp.sce.carleton.ca/pub/netmanage/integration.ps.gz.