Towards Distributed and Dynamic Network Management*

Akhil Sahai
Christine Morin
INRIA-IRISA
Campus de Beaulieu
35042, Rennes CEDEX, France
{asahai,cmorin}@irisa.fr

ABSTRACT: In this paper we describe a distributed and dynamic architecture for network management of a heterogeneous distributed system which we have implemented in order to perform network management of our local distributed system. The architecture is highly generic in nature and can be extended to a large heterogeneous network. The architecture is distributed in nature as it provides for multiple managers acting simultaneously. It is rendered dynamic by the utilization of mobile agents for the implementation of network management functionalities. The network managers in our architecture utilize client- server technology and/or mobile agent technology as and when required depending on the functionality implemented and their location. In this context we have implemented a Mobile Network Manager (MNM) which is a location independent network manager. Mobile agents are furnished by MAGENTA (Mobile AGENT environment for distributed Applications) which we have implemented for this purpose.

Keywords: Network management, mobile agents, mobile computing, Java.

1 Introduction

Network management comprises of network monitoring and control. A network management system thus performs among other things the tasks of configuration of the network, guaranteeing its proper functioning, maintenance, security control, collection and archiving of data and fault management. The prevalent network management systems in practice lack distribution, fault-tolerance and flexibility. The network management architectures have also not kept pace with the latest technological developments. The emergence of new technologies paves way for a rethink and change in the network management architectures.

We propose a distributed, dynamic and fault-tolerant architecture incorporating features of the centralized and fully distributed network management architectures. The architecture also incorporates the new technologies of Java

*This work has been partially funded by Dyade, R& D joint venture between BULL and INRIA

and mobile agents. In the proposed architecture multiple network managers operate simultaneously. The network managers are simple, light-weight and cost-effective. They are also system and location independent in nature. The manager being light-weight and portable can be executed from either a static computer or a mobile computer. In the architecture, mobile agents are used to decentralize as well as to distribute the network management functionalities in addition to the traditional client-server mechanism. They are also used to enable the functioning of the MNM. The mobile agents are furnished by the MAGENTA environment which we have designed and implemented in this regard.

Rest of the paper is structured as follow. The next section provides an overview of network management systems. Section 3 presents the architecture. The subsequent section describes the implementation details of the architecture before conclusion.

2 Overview of Network Management

A typical network management system comprises of one or more Network Management Stations (NMSs) which interact with the agents located at the network components utilizing a particular protocol (SNMP, CMIP, SNA etc.). The information communicated between the NMS and the agents is defined by a management Information base (MIB). A typical network management system is shown in Figure 1. It consists of GUI, management applications, NMS kernel and the database (DB). The NMS kernel obtains the management information from either the management agents or the DB and provides it to the management applications. The Network Management Station can be logically subdivided into a manager and a server. The manager comprises of the GUI and the applications which perform the network management activity, while the server comprises of the NMS kernel which obtains, saves and provides the information to the manager part of the NMS.

The prevalent network management architectures differ in the location and the amount of partitioning of the databases and the subdivision of the information gathering mechanisms amongst the NMSs. In this paper SNMP [1] is considered as the network management protocol for explaining in detail the network management functionalities as it is representative of network management protocols. The prevalent architectures of network management systems are depicted in Figure 2 and are described as follows:

Centralized network management. A single centralized NMS overlooks the management. It queries the network components on a timely basis to determine the health of the network. In this case the centralized Database (DB) stays with the NMS and it is the central NMS which communicates with the SNMP agents.

Hierarchical network management. A central NMS is aided by a set of subordinate NMSs. The subordinate NMS take off some of the responsibilities of the central NMS. Here the DB stays with the central NMS. The subordinate NMSs communicate with the SNMP agents and with the central NMS.

Peer network management. A set of NMSs manage the different domains of the network with timely interaction amongst them. The DB is partitioned and partially replicated between the peer NMSs. The peer NMSs communicate with the SNMP agents in their domains.

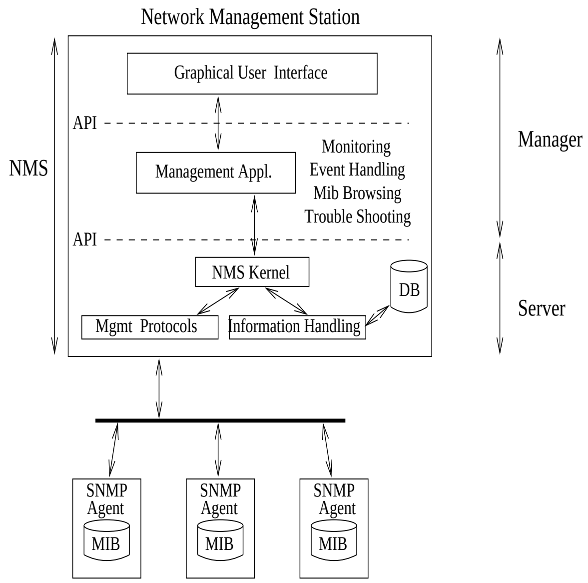

Figure 1: A typical Network Management System

Fully distributed network management. A totally distributed management architecture in which a large number of NMSs perform the management. The DB is highly partitioned and replicated in this case. The NMSs communicate with their assigned SNMP agents and amongst themselves.

3 The Distributed and Dynamic Architecture

3.1 Objective

The objective for the design and implementation of the architecture was to have a distributed, dynamic, fault-tolerant and flexible architecture incorporating and benefiting from the latest technologies. Latest trends in network management also has been towards decentralization and delegation [4]. An effort towards distribution and separation of GUI and applications from the information gathering part of a typical NMS has also been made [5][6]. Of late, mobile agent technology has generated lot of attention and has been applied in various domains. Mobile agents are autonomous programs which perform a goal-directed behaviour and have the capability of moving from one site to another, reacting to changes in the environment and interacting with the resources or other service/mobile agents at the sites [3]. Mobile agents can be utilized effectively for network management functionalities and for rendering the architecture distributed and dynamic.

3.2 Mobile Agents in Network Management?

Mobile agents can be used for a variety of purposes in network management. We utilize mobile agents for the following advantages they provide:

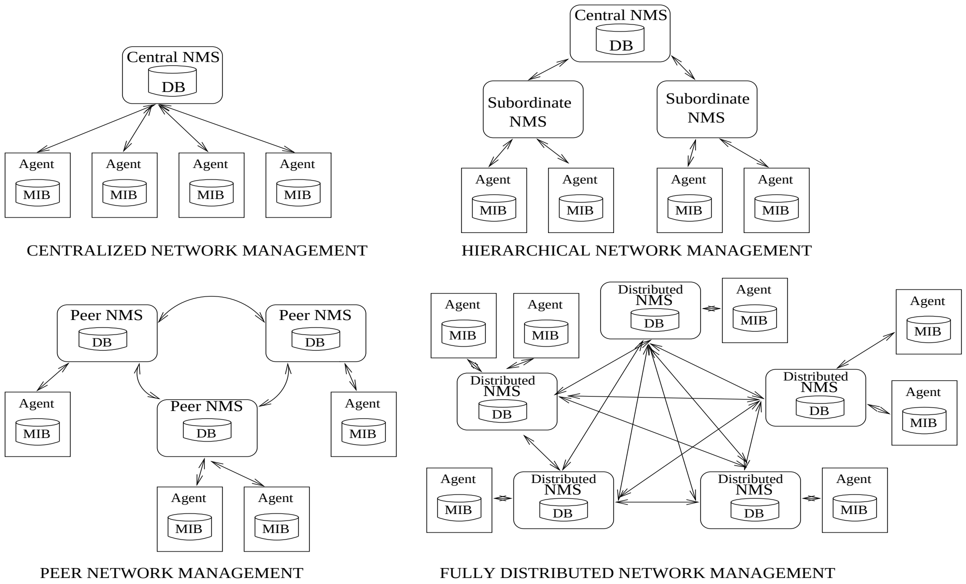

Figure 2: The prevalent architectures

Distribution of management code Mobile agents are used to distribute the code to the SNMP agents as and when necessary, instead of moving large amount of data to the manager over the network which reduces substantially network bandwidth usage and reduces the network bottleneck as well as makes the architecture more scalable(e.g in ATM switch management and Mib filtering the amount of data involved is substantial).

Decentralization They are effectively used to decentralize network management activities. They do not need continuous attention of the administrator in some of the network management activities e.g installation and upgrade of software, periodic audit of network.

Dynamic changing of network management policies. As the network environment grows and changes the policies need to be changed over time and also in order to tackle temporally changing problems like intrusion detection the management policies needs to be altered. Instead of going through rewrite, compile and run cycle the management policies are dynamically changed by writing new agents easily without altering the provided infrastructure.

Monitoring and statistics Mobile agents are suitable for retrieving large number of samples of SNMP variables i.e. suitable for monitoring of SNMP variables and also for studying the behaviour of network components over long period of time. In our architecture they are used for network components monitoring and for gathering statistics.

Data Collection They are suitable for data collection, searching and filtering. Thus they are used to query the servers for large quanta of data.

High speed networks They are suitable for high speed network management(Gb/s) in which case it is not practical to bring all the network data to the manager.

Disconnected mode of computing They are suitable for disconnected mode of computing and specially so in the case of mobile computing they reduce the usage of a fallible and costly link. They have been used to thus enable the functioning of the Mobile Network manager (MNM).

3.3 The Architecture

The proposed architecture is highly modular in nature. It comprises of managers, servers and management agents as shown in Figure 3. The manager consists of the GUI and the application part of a typical NMS, while the server consists of the NMS kernel, the DB, the protocol and mechanisms for communicating with the management agents. There is thus a clear separation between the application and the data collection part of the NMS amongst the manager and server in our architecture. The manager are architecturally neutral, light-weight and simple in nature. The managers can also access the servers simultaneously. This enables functioning of multiple managers. Thus we obtain the benefit of highly distributed architectures. However the servers are limited in number and the number of servers is decided by the size of the network. The servers have replicated DBs. So, the information gathering and storage is centralized in nature and we derive the benefits of centralized architecture. The network management load is thus equally distributed amongst the managers and the servers. Since the managers and servers typically execute at different sites there is thus a distribution of load. The managers are capable of obtaining the information from either the servers or the management agents thus there is no bottleneck at the servers and not all requests and data are channelized through the servers. This also further reduces the load of servers. The servers store only the information which needs to be archived and comprises of various daemons which continuously collect information about the configuration changes in the network, perform network discovery etc.

The managers, servers and the management agents are equipped with communication mechanisms to interact amongst themselves. Each manager is assigned a server however it can connect to other servers in case the assigned server fails. This communication mechanism consists of client-server as well as mobile agent paradigm. The communication mechanism is chosen depending on the location of the manager and the suitability of the communication mechanism for the functionality being implemented as shown in Figure 4. The architecture retains the client-server mechanism for interaction between the components. As is the case in traditional network management systems the managers can communicate with the management agents through primitives. In our case, the managers communicate with the SNMP agents using *get*, *set* and *get-next* primitives. They also interact with the servers through client-server mechanism. The managers, servers and the management agents are integrated with the MAGENTA environment. The MAGENTA environment provides them the capability of sending, receiving and storing mobile agents. The architecture is rendered more distributed and dynamic by mobile agents which distribute management code to remote sites and enable dynamic changing of network management policies.

The managers in our architecture are simple and light-weight in nature as they comprise of only the GUI and the management applications part of a typical NMS. They are also meant to be architecture- neutral and hence system-

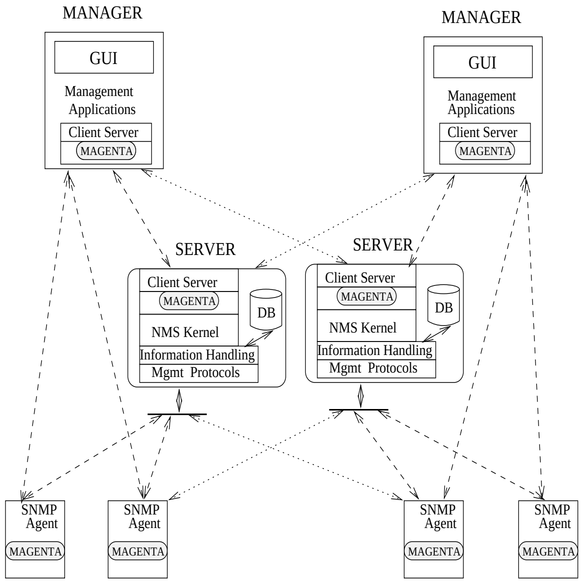

Figure 3: The distributed and dynamic architecture

independent. They are also dual-mode in nature i.e. they can execute on a static computer or a mobile computer. The manager operating on a mobile computer is thus location independent and is termed a Mobile Network Manager (MNM). The utilization of MNM can be envisaged in cases of emergency when the administrator needs to connect remotely to undertake management without loosing precious time. Other situations which could warrant the usage of MNM are when the administrator moves away from the network management station for some reasons e.g. software and hardware upgrade, meetings, vacations etc.

3.4 Advantages of the Architecture

The architecture provides for simple, light-weight, portable and cost-effective managers. It also decentralizes management logically as well as physically. Logically, because the network management information can be provided easily and at a much lesser cost to the users of the network and thus may help enhance the performance of the network. In addition since the architecture is distributed there is no need to dedicate any machine for network management.

Our architecture compares well with the prevalent architectures. It in fact benefits from the centralized as well as distributed architectures. As compared to distributed architecture, we have much lesser synchronization cost as the number of DBs is limited. In peer architecture, since the peer managers oversee the functioning of their local domains they usually maintain the information about their domains and information about other domains are obtained through request to peer managers. In our case, since all the information is conserved in the DBs, the information is available to the managers. We also have lesser synchronization cost and higher fault-tolerance as the architecture can continue to function inspite of failure of multiple managers, servers and SNMP agents. As

Functionality	Static Mode		Mobile Mode	
Technology used	Client Server	Mobile Agent	Client Server	Mobile Agent
Monitoring	✓		✓	✓
Performance	✓		✓	✓
Alarms	✓			✓
Correlation	✓			✓
Install		✓		✓
Audit		✓		✓
Network Health		✓		✓
Security of the Net		✓		✓
FTP		✓		✓
SNMP Study		✓		✓
Alarm Correlation		✓		✓
Fault management		✓		✓

Figure 4: Implementation of functionalities

compared to hierarchical architecture, it is much more fault-tolerant as the hierarchical architecture has a single point of failure while in the proposed architecture, the managers can continue functioning as long as it is able to access a single agent. The managers being equally intelligent and highly numerous the fault-tolerance can be ensured. The information gathering being semi-centralized in the proposed architecture is highly beneficial. Our architecture also compares well with the centralized architecture. The proposed architecture is much more fault-tolerant, decentralized, has multiple managers, is more scalable, and does not create a network bottleneck.

4 Implementation of the Architecture

The above architecture has been implemented and tested. Most of the network management functionalities are in place.

4.1 Environment

The architecture has been designed and implemented to manage the Astrolab distributed system[2]. The Astrolab distributed system comprises of a variety of machines running different operating systems like PCs (executing Win3.1, Win 95, Win NT, Linux, Net BSD), Sun Sparcs and AIX machines executing many variants of UNIX environment. The environment being heterogeneous in nature it was necessary to have a system independent manager. The managers have been thus written in Java which interacts with a server utilizing an infrastructure furnished by ISM/OpenMaster[7]. The managers, servers and SNMP agents utilize the MAGENTA environment which is also written in Java. For the implementation purposes Java was chosen because it is architecturally neutral.

It is object oriented, dynamically extensible and provides facilities of Object Serialization which was used for implementing mobile agents. Java also provides the capability of native methods to access the local operating system and thus is utilized for obtaining system information.

4.2 MAGENTA

In the proposed architecture mobile agents play an important role. The mobile agents are furnished by MAGENTA which is a generic environment for execution of mobile agents. It essentially comprises of *lieus* and *agents* as shown in Figure 5. A lieu is a static program capable of creating, sending, receiving and saving an agent and has a globally unique name. While the agents are autonomous programs which roam amongst the lieus interacting with the system resources, lieus and agents if necessary. The agents have a globally unique name, predefined purpose, a permit for execution, and a folder to carry their results. The agents are allowed to access the services through the service access point only after checking their permits and the address of the sending lieu.

MAGENTA provides for migration and remote execution. The agent can be either remotely created, executed and their results collected or can be moved from one lieu to another dynamically along with their state. The MAGENTA agents communicate with other agents on a lieu by exchanging messages called notes. The environment is flexible as the agents have the choice of either bringing the executable code with them or of executing code present at the site of execution. Mobile agents are prone to disappearance because of very nature of their execution and that of their itinerary which involves usually multiple machines. In MAGENTA backup copies of the agents are maintained so as to recreate agents which disappear due to site failures. A Directory mechanism is also provided to trace the agents. The dynamic disappearance and appearance of lieus is also taken into account. The lieus can disappear gracefully through an exit mechanism or can disappear abruptly. In the latter case agents which discover such disappearances inform other lieus of the non-existence of the disappeared lieu.

4.3 Functioning of the Architecture

The managers, servers and SNMP agents of the architecture are integrated with MAGENTA lieus. The dumb devices which can not support a lieu are managed directly by managers using client-server mechanism. Thus the mobile agents can be sent, received, executed and saved at any of the network element supporting a lieu. The manager operating either in static mode or in mobile mode can send agents for performing the required functions.

In a static mode as shown in Figure 6, the manager in the wired network can send an agent to a server and/or to SNMP agents and perform a network management function. It can also obtain the information from the SNMP agents and the servers using a client-server approach depending on the functionality implemented.

In the mobile mode the manager (MNM) can operate either in tethered manner or in a wireless manner. In tethered manner the MNM sets up a PPP or a SLIP connection to the intranet, launches the agents to perform the desired action and tears down the connection. The MNM reconnects back using similar

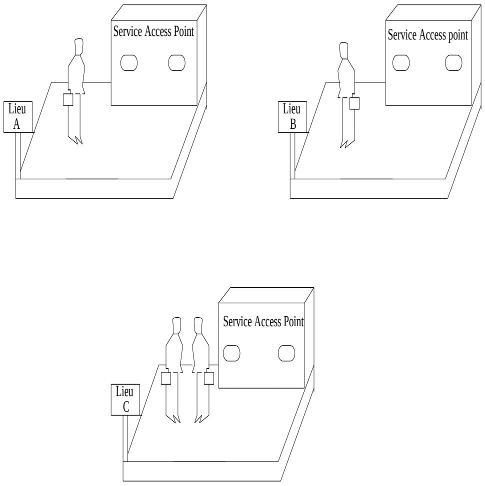

Figure 5: MAGENTA:lieus and agents

protocol and collects the agents waiting for it in the wired network which come back to the MNM on its reconnection. In case of a wireless MNM as shown in Figure 6, as the manager moves from one cell to another, the mobile agents passes through the Mobile Service Station (MSS) of the previous cell and after completing its itinerary comes back through the new MSS of the new cell [8]. In case the MNM disconnects after sending the agent, the mobile agent waits at the last lieu in the itinerary for the MNM to connect back. As soon as it connects back, the agent returns with the results of the computations performed by it.

5 Conclusion

In this paper we have proposed a highly fault-tolerant, distributed and dynamic network management architecture which draws on strengths of centralized and distributed architectures. The network management architecture uses the client-server communication model or the mobile agent communication model depending on the functionality implemented and on its mode of operation. The architecture also provides for system independent and location independent manager. The architecture has been fully implemented and a prototype destined for managing large intranets has been developed.

The architecture incorporates mobile agents for implementing network management functionalities. However, the domain of mobile agent computing is relatively new and is undergoing development. Solutions to all possible kinds of security aspects of the mobile agents are being studied. This restricts the usage of mobile agents to large and secure intranets. However solutions to the security aspects will lead to their utilization in large networks made up of multiple intranets.

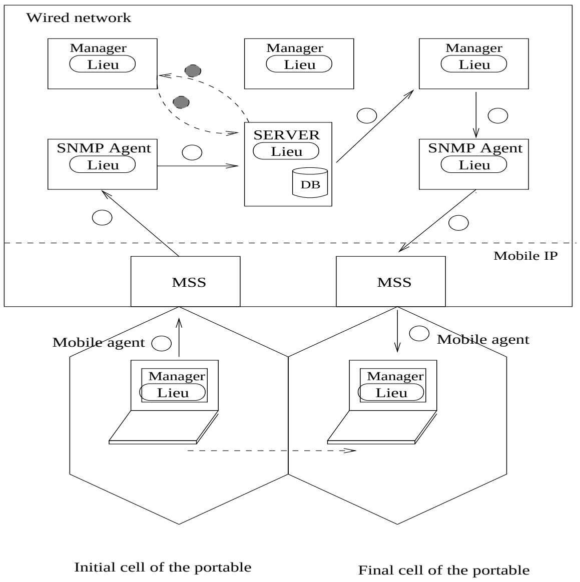

Figure 6: Functioning of the architecture

References

[1] W. Stallings. SNMP, SNMPv2 and CMIP: The practical guide to network management standards. *Addison-wesley publication, 1994.*

[2] Sahai A., Billiart S. and Morin, C. Astrolog: A distributed and dynamic environment for network and system management.*In the Proc. of 1st European Information Infrastructure User Conference, Stuttgart, Germany, Feb. 1997*, http://www.irisa.fr/ solidor/work/astrolog.html.

[3] Wooldridge M. and Jennings R. N. Agent Theories, Architectures, and Languages: A Survey. *Springer-Verlag Lecture Notes in Artifical Intelligence 890*, pp 1-39

[4] Goldzmith G. and Yemini, Y. Decentralizing Control and Intelligence in Network Management. *In the Proc. of 4th International Symposium on Integrated Network Management, Santa Barbara, CA, 1995.*

[5] Advent Net Monitor fabricated by Advent Network Management Inc. *http://www.adventnet.com.*

[6] Java Management API by Sun MicroSystems. *http://java.sun.com/products/JavaManagement/.*

[7] Integrated System Management/ Open Master by BULL. *http://www.ism.bull.net/.*

[8] Bakre A. and Badrinath B.R. I-TCP: Indirect TCP for Mobile Hosts *Technical Report DCS-TR-314, Department of Computer Science, Rutgers University, 1994.*