Astrolog: A Distributed and Dynamic Environment for Network and System Management

Akhil *Sahai*†, Stéphane *Billiart*‡, Christine *Morin*†
† INRIA
‡ BULL
IRISA, Campus Universitaire de Beaulieu
35042 Rennes Cedex (France)
{asahai, billiart, cmorin}@irisa.fr

Abstract

Astrolog[1] is a highly dynamic and decentralized management system meant for the management of Astrolab, which is a distributed system comprising of heterogeneous machines running varied operating systems connected by a LAN. The design of Astrolog strikes a balance between a centralized management system and a totally distributed management system drawing on strengths of both the mechanisms. It introduces a unique dynamically variant hierarchical management architecture. It also introduces the concept of a mobile network manager for the first time. Astrolog is designed for management of a platform which is in extensive and regular usage, thus the design is highly practical in approach.

Keywords: Network Management, System Management, SNMP, Mobile Agents, Java.

1 Introduction

Astrolog introduces a unique dynamically variant management architecture. It introduces the concept of a light-weight and cost-effective management environment which is highly portable and at the same time highly dynamic in nature. The concept of mobile network manager i.e. a management environment running on a portable computer is unique to Astrolog. It is being tried for the first time to provide a management environment on a portable which is highly restricted because of the lack of bandwidth available and because of lack of processing power. The mobile agent technology will be utilized to overcome the resource restrictions of a portable computer. This paper describes the ongoing research and development of Astrolog and its contributions to the area of system and network management.

[1]Part of this work is being carried out under the GIE-DYADE collaboration between INRIA and BULL.

2 Overview of Network Management Systems

Network-management functions can be grouped into two categories: network monitoring and network control. Network monitoring is a "passive function". It thus involves minimal interference in the status of the network, while network control is an "active function" and involves direct and active participation of the network management system.

The network-monitoring portion of network-management is concerned with observing and analyzing the status of the network to be managed. Network monitoring is an essential aspect of automated network management. The information to be gathered includes static information, related to the configuration, dynamic information related to events in the network and statistical information, summarized from dynamic information. It also involves identification of faults and determining the reasons of the fault and to take remedial actions. It also involves proactive response to an impending fault and minimization and containment of the fault.

Network control is concerned with changing the variable values of various components of the network and causing those components to perform predefined actions. The area of configuration control encompasses a variety of functions relating to the configuration of network and computing elements. These include initialization, maintenance, and shutdown of individual components and logical subsystems. In the area of security control, the responsibility of the network management system is to coordinate and control the security mechanisms built into the configuration of networks and systems under its management control. These security mechanisms are intended to protect user and system resources, including the network management system itself.

A network management system contains four types of components: Network Management Stations (NMSs), agents running on managed nodes, management protocols, and management information. An NMS uses the management protocol to communicate with agents running on the managed nodes. The information communicated between the NMS and agents is defined by a Management Information Base (MIB). The management standards that have emerged are the Simple Network Management Protocol (SNMP) and the OSI management system which utilizes the Common Management Information Protocol (CMIP)[5]. SNMP is simpler and more concise in comparison to CMIP which is more elaborate and provides many more functionalities. Astrolog utilizes SNMP in keeping with its intention of being light-weight and simple. The SNMP protocol includes the following capabilities mainly,

- *Get*: enables the management station to retrieve the value of objects from the agent;
- *Set*: enables the management station to set the value of objects at the agent;
- *Trap*: enables an agent to notify the management station of significant events.

The prevalent architectures of network management systems are

- Centralized network management. A single centralized manager overlooks the management. It queries the network components on a timely basis to determine the health of the network.
- Hierarchical network management. A central manager is aided by a set of subordinate managers. The subordinate managers take off some of the responsibilities of the central manager.
- Peer network management. A set of Network Managers manage the different domains of the network with timely interaction amongst them.
- Fully distributed network management. A totally distributed management architecture in which every agent shares the responsibility of management.

While centralized network management architecture and hierarchical management architecture have found extensive practical usage with great success, the peer management architecture and the fully distributed network management architecture have not been extensively used in practice. However, both the schemes being advantageous in different circumstances, we have decided to utilize a dynamic architecture. Astrolog thus introduces a architecture which alternates between a peer management architecture and a hierarchical management architecture drawing on strengths of both the architectures.

3 Astrolog

3.1 Context

Astrolog is designed to manage our local research platform called Astrolab which comprises of heterogeneous machines running several different operating systems and connected by a LAN. The machines are PCs(running Win95, WinNT, Linux, NetBSD etc), SUN Solaris workstations, AIX Workstations etc. Thus for us it was important to have an architecturally neutral management environment which is light-weight and cost-effective and at the same time highly portable. Astrolog is utilized for the management of our local distributed system Astrolab which is relatively smaller in size, and also in keeping with the intention of making Astrolog simple and light-weight we utilize only SNMP as the network management protocol. The idea was to make a management environment for our local system and also to introduce new ideas in the realm of network management like that of dynamically variant management architecture and of mobile network manager.

3.2 Design of Astrolog

In Astrolog the managers are light-weight, cost-effective but quite elaborate in their functionality. These managers are also designed to be portable and can run from a variety of machines over the network in a client-server mode or in the mobile agent mode or sometimes utilizing both. The managers are Java applications which have the functionality of network monitoring and control. The managers are served by servers. The servers run on one or more sites containing the databases depending on the size of the network. These databases containing the information about the network are populated by respective daemons executing on the sites. The servers obtain the required information from the database or from the agents located at the managed systems and provide the information to the managers as and when requested by them. At any instant a server can serve more than one manager. However, in some cases mobile agents are used for communication between the managers and servers most notably for information retrieval and to overcome the constraints of the network and the processing power as in the case of a mobile computer. In contrast with a typical centralized management system in which the single centralized manager comprises of the GUI, the management applications and the NMS kernel along with the database as shown in Figure .1, Astrolog is divided into light-weight managers comprising of the GUI and the management applications like MIB browsing and monitoring and a mechanism of connecting to the server and obtaining the network management information. This part acts as the client. The other part comprises of the NMS kernel, the DB and a communication mechanism to interact with the client, which in turn acts as a server. There can be multiple managers querying one or more servers (depending on the size of the network). The managers (clients) can either exist on the same site as the server or can also exist at a different site and can query the server over the network as shown in Figure. 2.

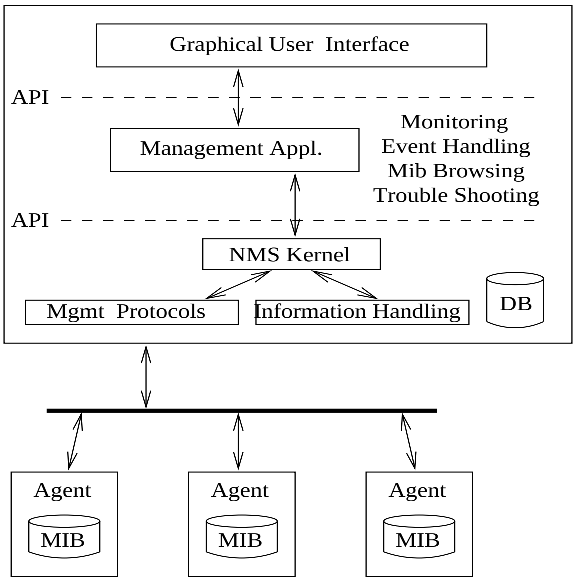

Figure 1: A typical centralized Network Management System

The design of Astrolog provides for multiple local managers. These equally intelligent multiple local managers function as domain managers and use SNMP for network management [2][3]. They also use meta-variables derived from normal SNMP variables to perform efficient management. In [9] the utization of meta-variables has been shown to be quite effective. Since the managers utilize the meta variables, they can monitor the network more efficiently. It can be shown through some examples that meta-variables are indeed the key to efficient management. Let us consider the following:

- Interface monitoring

  In most network installations, routers have an important position. The interface status needs

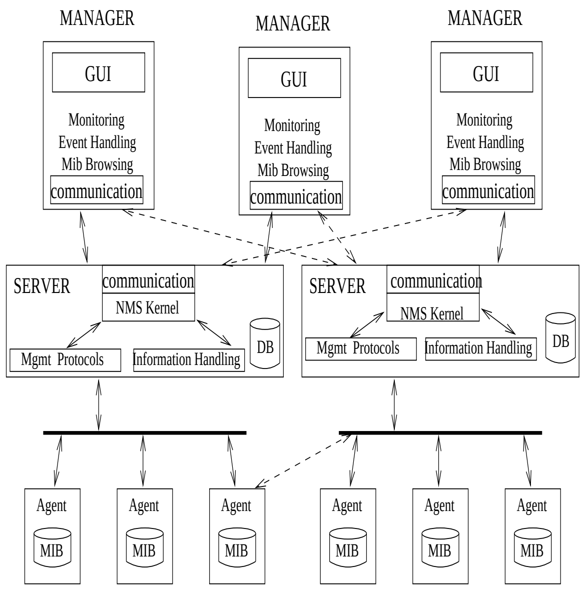

Figure 2: Design of Astrolog Management System

to be checked regularly. The *ifAdminStatus* and the *ifOperStatus* of the interfaces can be utilized to create high level meta variables.

- Rate monitoring

  For an IP entity total number of input datagrams received from interfaces including those received in error is given by *ipInReceives*, while the total number of input datagrams successfully delivered to the IP user protocols is *ipInDelivers*. By utilizing a simple error rate mechanism as follows

  $$result = \frac{(ipInReceives - ipInDelivers)}{ipInReceives}$$

  and by monitoring the change in the error rate any drastic losses because of network routing errors, lack of buffer space can be easily determined.

- Logical monitoring
  A set of network entities are treated logically as a group when they are physically placed together. Thus meta-variables can be created which gives the state of health of such groups. such information will be very helpful in case of breakdown, in which case the possible reason of failure can be determined by pinpointing the faulty clusters and identifying the possible cause of failure.

It is thus apparent that the meta-variables if used efficiently can lead to more useful management. Astrolog thus comprises of a group of domain managers which perform their operation similar to a peer management network. However, the architecture changes dynamically. In Astrolog, the management information is stored into fewest possible management databases so that it is more robust and benefits from centralization but it is also highly dynamic and decentralized with equally intelligent managers managing their local domains and utilizing mobile agents for management. An administrator specified update rate is utilized to update the SNMP [4] [5] management database(s) which stores the SNMP network variable instance values. The local managers perform the management of proximate devices. This thus leads to efficient management responsibility allocation. They access the management database(s) to access the latest SNMP variable instance values. The local managers are thus capable of displaying the overall present network and system information on demand by the system administrator(s) or the system users.

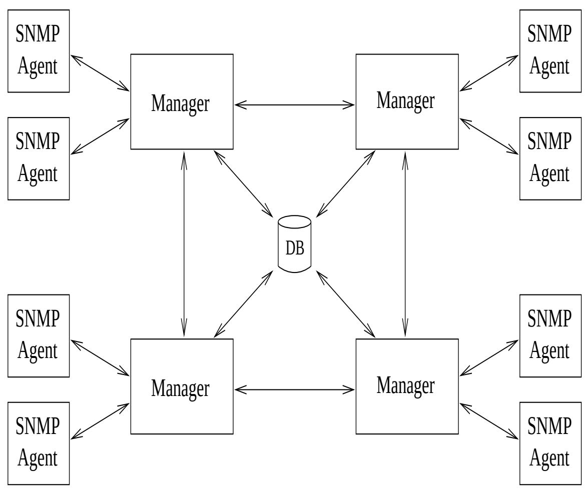

Figure 3: Astrolog in normal operation

The domain managers are equally intelligent and obtain the values of the network management components from the database. The domain managers normally obtain the values over the network by utilizing a client-server paradigm. However, there is a facility in Astrolog of obtaining the values utilizing mobile agents in case of a direct link failure in which case mobile agents try to obtain the values from the peer managers. The mobile agents will be also used when the amount of data to be read is quite huge and it is impractical to do it over the network.

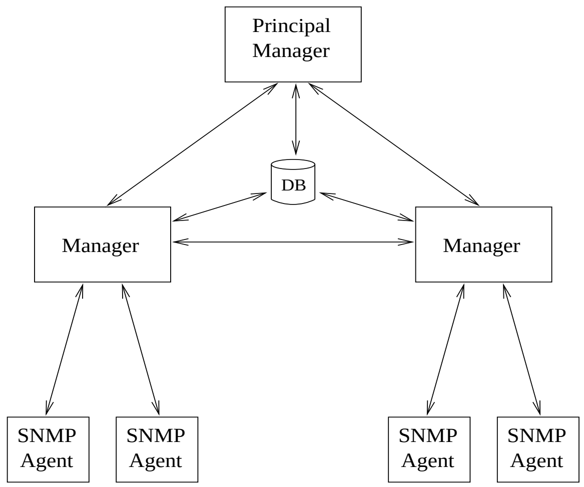

Figure 4: Astrolog during a crisis management operation

During a crisis when a system administrator logs into one of the network managers, that particular network manager is termed as the principal manager and the other network managers act as subordinates to this particular manager. Thus the architecture changes to a hierarchically arranged management architecture. The scheme thus draws on the strengths of both, namely a peer management architecture and a hierarchical management architecture. Further, a portable manager lends additional flexibility to network management. It also enables utilization of larger number of managers thus providing the users an opportunity to observe the state and the performance of the distributed system.

What facilitates the flexibility of the architecture is that the network managers are light-weight and thus there can be multiple interchangeable managers. We are utilizing this light-weightedness of the manager to build mobile network managers. Nowadays in case of a crisis the system administrator is informed by the management system through a pager message. one of the recourse left to the administrator is to rush to the central management station. In case of the absence of the system administrator in the proximity of the centralized management station the system administrator is not aware of the exact nature and cause of the crisis and is thus unable to undertake important decisions in case of a network breakdown. In case of a large network there can be more than one administrator and it is sometimes essential to seek the opinion of more than one administrators at the same time. Also there is an increasing trend towards wireless networks and it will thus have a wireless network manager thus we consider that the need and usefulness of a mobile network managers are immense. Mobile network managers are thus managers that run on portable computers which can be running in the tethered mode using a PPP/SLIP mechanism or can be a roaming wireless computer. Our design takes care of both the situations.

What distinguishes these mobile computers are the tremendous constraints on the link available to them. The links have serious bandwidth constraints, have high latency and are prone to sudden failures such as when a signal from a cellular modem is blocked by an obstacle. The computer may be forced to use different transmission channels depending on its physical location. Finally depending on the nature of the transmission channel, the computer may be assigned a different network address each time it connects. Both wireless networks and phone lines are orders of magnitude more constrained than traditional LANs [8] as shown in tables 1 and 2. Table. 1 shows the dramatic discrepancies between bandwidth (kilobytes per second) and the network round trip times (latency) of the media:

Networks	BW(kbps)	Latency
LAN	5K-10K	.0005-.001 secs
Modem	2.4K-28.8	0.2 - 0.5 secs
Wireless WAN	2K-9	4-10 secs

Table 1: Comparison of Networks ([8])

Normally a client-server paradigm is utilized for most of the distributed computation. A client-server order status application might exchange fifty criteria about the retrieval of multiple rows of data. Table. 2 shows how that application performs over a LAN, a phone line, and a wireless network.

Network	Ass.Latency	Min Resp.Time
LAN	0.002 secs	0.1 secs
Modem	0.3 secs	15 secs
Wireless WAN	4 secs	200 secs

Table 2: Comparison for 50 round trips ([8])

Therefore client-server paradigm is not very useful in the case of a partially connected computer and hence for mobile network manager. We utilize the mobile agent technology to take care of this problem.

For implementing this scheme of mobile network managers, we model an indirect interaction. There is thus a concept of a static proxy in our design. One of the existing static managers is allocated to each portable computer to act as a proxy. The proxy acts as an intermediary between the portable computer and the server. The agent emanating from the portable computer on reaching the proxy convey the request to the proxy. The proxy tries to get the information from the server. In case the operation is successfully carried out, the agent before going back to the portable computer check whether it is still connected. If it is connected, the agent goes back to the portable computer otherwise the agent waits for the portable computer to connect back. The agents thus reduce the continuous usage of the highly constrained link and also take care of it fallibility. The tethered mode of operation is straightforward as depicted in Figure. 5. In the tethered mode the portable is connected to the static network through a PPP/SLIP mechanism and the mobile manager runs on the portable which sends mobile agents to retrieve the required information from time to time. These agents collect the information and either return back immediately if the portable remains connected otherwise they wait at the proxy and return back as soon as the portable is connected back.

The wireless mode of operation needs to be explained in detail. Normally in the area of wireless computing the total domain is divided into cells.

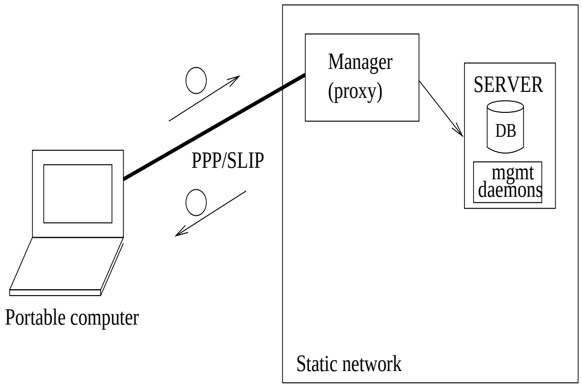

Figure 5: Portable network manager in the tethered mode

Each of the cells have a Mobile Support Station (MSS) of their own. The responsibility of serving the Mobile Host (MH) as it moves from one cell to another cell changes from one MSS to another [7]. The utilization of a static proxy serves us in good stead. The mobile agent as launched by the portable computer (MH) go to the proxy which in turn obtains the required information. In the meanwhile if the MH moves to a new cell the new MSS is informed by the MH to retrieve the agent waiting at the proxy with the results. The proxy then delivers the agent at the new location of MH as shown in the Figure .6

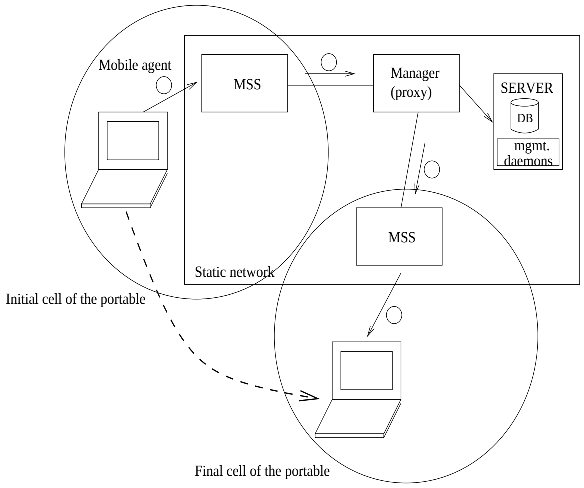

Figure 6: Portable network manager in the wireless mode

4 Implementation

The server part comprises of a database, a communication mechanism, a network management system kernel which also comprises of discovery daemon. The discovery daemon discovers the devices and populates the database. The NMS kernel also comprises of daemon which queries the SNMP capable devices to know their status and stores the information in the database. The manager comprises of the GUI, applications and a mechanism to access the information from the server. The manager has various network management applications to determine the performance, alarms, to do monitoring and mib browsing etc.

Java [6] was chosen for implementing the local managers because it is architecturally Peutral. It was intended to have multiple managers accessing the server from a variety of plat forms, thus only the local managers needed to be platform-independent. The platform independence was a necessity in our case because Astrolab is a heterogeneous system. Java is also object oriented and dynamically extensible and thus is suitable for writing mobile agents. Java also provides the capability of native methods to access the local operating system and thus is also suitable for obtaining system information. Java also provides the capability of deriving meaningful meta-variables from the available SNMP variables to the local managers enabling them to monitor the local domains more effectively. The local managers are thus capable of providing the system users and the system administrator an easily portable network and system visualization and management editor. The GUI part of the manager comprises of various Java classes which have been written by us and the network management applications utilize the GUI classes to display and depict the accessed information. These classes provide the facility of depicting the available information in the form of pie-charts, line-charts, histograms and line diagrams. A large number of classes for creating dialog boxes, message-boxes , hierarchical lists, tables, scrolling windows, image buttons and for drawing topology have also been implemented.

Network management information is accessed from the management database by a high-level protocol or directly from the agents located at the managed systems utilizing SNMP. The mobile agents, which will be launched from the managers either in the mobile mode or in the static mode are being implemented. These mobile agents are being written in Java and they utilize Java object serialization. A set of agent functionality has been defined and the agents are being implemented in a generalized fashion so that the same agents can be utilized in different contexts like that of electronic commerce.

5 Conclusion

Astrolog contributes significantly to the ideas of network management. It not only introduces the unique dynamically variant network management architecture but also introduces the concept of a light-weight, architecturally neutral and a cost-effective manager. It also introduces the unique concept of a mobile network manager.

6 Acknowledgments

We would like to thank Serge Lassabe for the discussions we had with him and for his invaluable suggestions. We would also like to thank Janusz Wisniewski for assisting in the development of the Astrolog platform.

References

[1] German Goldzmith. Management By Delegation *Technical Report, Columbia University 1996.*

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

[3] J. Case, M. Fedor, M. Schoffstall, J. Davin. A Simple Network Management Protocol (SNMP). *RFC 1157.*

[4] M. Rose, K. McClogherie. Structure and Identification of Management Information for TCP/IP-based internets (SMI). *RFC 1155*

[5] U. Warrier, L. Besaw. The Common Management Information Services and Protocols over TCP/IP (CMOT).*RFC 1095*

[6] James Gosling and Henry McGilton. The Java language Environment : A white paper. *Technical Report, Sun Microsystems, 1995.*

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

[8] Oracle White Paper: Oracle Mobile Agents. *Technical report August 1995.*

[9] Manfred R. Siegl and Georg Trausmuth. Hierarchical Network Management : A concept and its prototype in SNMPv2 .*Proceedings JENC6, 1995*