Adaptive management of emerging battlefield network

Fountoukidis, Dimitrios P.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ADAPTIVE MANAGEMENT OF EMERGING BATTLEFIELD NETWORK

by

Dimitrios P. Fountoukidis

March 2004

Thesis Advisor:	Alex Bordetsky
Thesis Co-Advisor:	John Hiles

This Thesis is done with cooperation with MOVES Institute

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Adaptive Management of Emerging Battlefield Network			**5. FUNDING NUMBERS**
6. AUTHOR(S) Dimitrios P. Fountoukidis			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT

The management of the battlefield network takes place in a Network Operations Center (NOC). The manager, based on the importance of the managed network, is sometimes required to be present all the time within the physical installations of the NOC. The decisions regard a wide spectrum of network configurations, fault detection and repair, and network performance improvement. Especially in the case of the battlefield network operations these decisions are sometimes so important that can be characterized as critical to the success of the whole military operation. Most of the times, the response time is so restricted that exceeds the mean physical human response limits. An automated response that also carries the characteristics of human intelligence is needed to overcome the restrictions the human nature of an administrator imposes.

The research will establish the proper computer network management architecture for an adaptive network. This architecture will enhance the capabilities of network management and in terms of cost and efficiency.

14. SUBJECT TERMS: Adaptive Network, SNMP, Mobile Agents, Artificial Intelligence, Collaborative Agents, MANTRIP Project.			**15. NUMBER OF PAGES** 114
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

ADAPTIVE MANAGEMENT OF EMERGING BATTLEFIELD NETWORK

Dimitrios Fountoukidis
Lieutenant Commander, Hellenic Navy
B.S., Hellenic Naval Academy, 1989

Submitted in partial fulfillment of the
Requirements for the degrees of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

MASTER OF SCIENCE IN MODELING VIRTUAL ENVIRONMENT AND SIMULATION

from the

NAVAL POSTGRADUATE SCHOOL
March 2004

Author:	Dimitrios Fountoukidis
Approved by:	Dr. Alex Bordetsky Thesis Advisor
	John Hiles Co-Advisor
	Dr. Dan Boger Chairman, Department of Information Sciences
	Dr. Rudy Darken Chairman, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The management of the battlefield network takes place in a Network Operations Center (NOC). The manager, based on the importance of the managed network, is sometimes required to be within the physical installations of the NOC all the time. The decision makers must consider a wide spectrum of network configurations, fault detection and repair, and network performance improvement. Especially in the case of the battlefield network operations, these decisions are sometimes so important that they can be characterized as critical to the success of the whole military operation. Most of the times, the response time is so restricted that it exceeds the ability for humans to respond. An automated response that also possesses the characteristics of human intelligence is needed to overcome the restrictions that a human nature of an administrator imposes.

The research will establish the proper computer network management architecture for an adaptive network. This architecture will enhance the capabilities of network management and in terms of cost and efficiency.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to acknowledge the Hellenic Navy for the opportunity that they have given me to learn and further my education in a conducive environment like – The Naval Postgraduate School.

I am also thankful to Dr. Alex Bordetsky and John Hiles for their constant willingness to teach and advise me. Their guidance and enthusiasm has helped me to complete this project.

I would also like to acknowledge the contribution and help that Don McGregor, Eugene Bourakov, and Lieutenant Commander Chris Eagle USN rendered in various and sometimes critical parts of my project.

I am most grateful to my wife and daughter, Valentina and Matina, whose love, understanding and support were essential during the development and writing of this thesis. Their patience throughout my course of study is truly appreciated.

I would like also to honor and thank my parents, Panagiotis and Stamatoula, for their love, sacrifice, and for establishing the sound and correct foundations upon which my character has been built.

Last but not least, I thank God, for his grace, without which I would be lost and would not have this opportunity to do what I have accomplished.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

This thesis concerns the Adaptive Management of the emerging Battlefield Network. The first step was to explore the capabilities of the legacy SNMP protocol regarding the control of network components and network links.

For this purpose we established a small network comprising of a manager station and three managed clients. We tried to alter the configuration of the links among the clients using plain SET SNMP instruction through the manager station. The results show that the SNMP protocol has inherent design deficiencies regarding this operation, and it was apparent that we could not control the QoS attributes of the Network using the SNMP protocol.

Next, we proposed a new Adaptive Management Protocol that could substitute the legacy SNMP and provide new capabilities to the network manager. In this way, we could construct an Adaptive Network Stack by superimposing an Artificial Intelligence Layer that could substitute for the Network Administrator – Decision Maker.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BATTLEFIELD NETWORK MANAGEMENT TECHNIQUES

Information superiority and the advances in the field of computer network technology are transforming the traditional battlefield. We are experiencing a new revolution in military affairs that takes advantage of the computer network technology in order to achieve the desired advantage in tactics but more significantly in strategy.

A battlefield network resembles an inter-network. Various technologies and protocols use and explore the TCP/IP stack in order to achieve a seamless connectivity. Satellite links are working together with wire or wireless links and this orchestra of networks tends to be rather unstable. Monitoring and control of the battlefield network is achieved by using Network Operation Centers (NOC). These physical installations are manned continuously during critical times of military operations. Network stability and availability is so critical in today's Network Centric Operations that minutes of an inoperative link can reverse the current of battle operations.

The operation of controlling and managing a battlefield network generally requires pre-emptive decisions that must be made in seconds and if possible in milliseconds. The flow of data from and to the network clients and servers is so overwhelming that the administrators rarely notice the real warnings. As the use and complexity of battlefield networks increase geometrically, there is a great need to automate processes and even automate the decision making.

The NOC collects, integrates, and displays data measurements taken from the underlying network. Let's consider, for example, a wireless "customer" with a cell phone/PDA that handles voice, e-mail, collaboration utility, and GPS. This individual needs to access cellular, fixed wireless and satellite networks, each with its own NOC. The question that instantly emerges is how we can manage customer service within this environment. Although the NOCs for each of these networks may have many similarities, they also have important differences. The goal of the network managers is to achieve a seamless integration of all these subsystems so their operation can be abstracted from the customer.

Regarding the management of the battlefield network, the decision making process takes place in the NOC. The decision maker, based on the importance of the managed network, is sometimes required to be present within the physical installations of the NOC all the times. His decisions regard a wide spectrum of network configurations. Especially in the case of the battlefield network operations, these decisions are sometimes so important that they can be characterized as critical to the success of the whole military operation. Generally, the response time is so limited that it exceeds the ability for humans to response. An automated response that also possesses the characteristics of human intelligence is needed to overcome the restrictions that the human nature of an administrator imposes.

Traditionally, connection-oriented network technologies (e.g. PDH, SDH, and ATM) have been the main focus of network management. Due to the recent use of IP backbones and emerging IP technologies that support QoS (e.g. IntServ, DiffServ, and MPLS), the network management area has been enlarged in scope. The use of IP technologies is growing; all-IP environments ranging from access to core-networks are increasing in the telecommunications world. The problem consists in dimensioning and correctly administering network resources according to the new service requirements.

IP network technology enters the telecommunication market due to both its simplicity and low price. It can be suitable for supporting various types of applications but it is not suitable for services with a high quality of service (e.g. real time audio and video services). The new IP technologies with QoS capabilities try to support these requirements but at the same time pose a new problem: effective network resource management. IP routers must be managed, that is, configured, monitored and dynamically reconfigured in order to meet user requests. Clients are becoming more and more demanding while services must be provided fast and executed efficiently. Therefore, operators and service providers should be prepared to respond to client requests quickly. Robust and flexible management platforms are key enabling factors to help providers in honoring their contracts (i.e. Service Level Agreements).

Existing network management systems are typically dedicated to managing of certain type of network resources. These are often vendor and technology specific and make use of centralized architectures, which are not flexible in principle. Management information can grow significantly and introduce substantial delays in the network. Scalability of network management systems is frequently a limitation, which determines their applicability in managing large distributed networks, such as IP-based Next Generation Networks (NGNs).

Recently, Mobile Agent Technology (MAT) has emerged as a promising solution towards implementing strategies that distribute and automate management tasks. Even though there are still several open research issues, mobile agent technology appears mature enough to support distributed and decentralized network management. Mobile agents can easily migrate to remote locations, execute their tasks and return with results. This capability allows agents to travel from one node/machine to another and perform a repetitive task or, when appropriate, several instances of the same agent can be created in order to perform the same task in parallel.

Standardization organizations made the effort of finding solutions for well-known problems such as security, portability, mobility, resource management and discovery. As a result, several mobile agent platforms are already available and new ones will become available in the future.

The research will establish the proper computer network management architecture for an adaptive network. This architecture will enhance the capabilities of network management and in terms of cost and efficiency.

B. SCOPE

Demonstration of the viability of the hypothesis that an integrated self managed and self-controlled network is possible based on Artificial Intelligence.

C. EXPECTED BENEFITS OF THE STUDY

This study will show the benefits of an integrated Computer Network Management System. In this system, the decision maker for the control and management is not a human but rather the system itself that will also develop an experience and become self-taught. A self-managed computer network is valuable not only to military installations but also to commercial applications in which the human resources for administering a network are very valuable and restricted.

D. OVERVIEW OF OTHER CHAPTERS

In Chapter II we cite the basic principles of Network Management and a description of the SNMP protocol. This will stand as the basis of our experiment that follows in Chapter III where we explore the capabilities of the SNMP protocol by setting up a small network and trying to control QoS attributes by using only SNMP instructions.

Chapter III describes the developed model for our experiment, the Microsoft Windows XP implementation of the SNMP protocol, as this is the main operating system that we employ, and the AdventNet Java API that we used to implement our model.

Chapter IV presents the principles of Artificial Intelligence and especially the tools that we use to build our Adaptive Network Architecture. These tools include AI agents, Multi Agent Simulation, Agent Communication, Case-Base Reasoning, and the deployment of Mobile Agents through a network.

Chapter V goes through the Adaptive Network Architecture by first introducing the MANTRIP project that stands as the starting point for our proposal. Next we introduce a new Adaptive Management Protocol that could replace the legacy SNMP protocol and provide new enhanced capabilities to network management. Lastly, we introduce an Integrated Adaptive Network Architecture for a completely human independent adaptive network.

Appendix I gives the basics on how we can enable the SNMP Agent in Windows XP platforms and Appendix II cites the code used to test the SNMP capabilities of the established experimental network.

II. NETWORK MANAGEMENT

A. INTRODUCTION

Network management can be defined as **OAM&P** (operations, administration, maintenance and provisioning) of network and services (Subramanian, 2000). Although the term "network" seems restrictive, it is very convenient to use, as the backbone for all the Information Technology services is still the computer network, regardless of its new and exciting morphs.

The Operations group is the main pillar of a network as it is occupied with the "up and running" aspect of the network. Although the rest of the management operations are more in support of the operations group, they are still of great importance, and they cannot in any way be overlooked. The Management administration ensures not only the enforcement of the appropriate policies but also the overall goals of the managers. The maintenance group is occupied with installation and repairs of the facilities. Provisioning takes care of the hardware support and the overall efficiency of the installation.

B. NETWORK MANAGEMENT

In the rapid pace of growing technology, networks tend to be large and complex, filled with many types of equipment from different vendors. Managing such a network is increasingly more difficult, involving multiple management tools and protocols to support different proprietary devices on the network. The goal of network management is to ensure that users of a network receive the information technology services with the quality service that they expect (Subramanian, 2000). Therefore, the network administrator needs a network management tool that can monitor the network's availability, utility and performance with the most industry-acceptable standards as possible to reduce the conflict of the network management system.

The purpose of network monitoring is to gather information about the status and behavior of network elements. 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.

Typically, each managed device in the network includes an agent module responsible for collecting local management information and transmitting it to one or more management stations. Each management station includes network management application software plus software to communicate with agents. The Information may be collected actively, by means of polling by the management station, or passively, by means of event reporting by the agents.

The International Organization for Standardization (ISO) Network Management Forum has divided network management into five functional areas:

1. Fault Management

The objective of fault monitoring is to identify faults as quickly as possible after they occur and to identify the cause of the fault so that remedial action may be taken. Comprehensive fault management is the most important task in network management.

Fault management tools can help increase the reliability of the network by quickly identifying the fault, isolating the cause of the fault, and then, if possible, correcting the fault.

2. Configuration Management

Configuration management deals with the initialization, modification, and shutdown of a network. Networks are continually adjusted when devices are added, removed, reconfigured, or updated. The process of configuration management involves identifying the network components and their connections, collecting each device's configuration information, and defining the relationship between network components. In order to perform these tasks, the network manager needs topological information about the network, device configuration information, and control of the network component.

3. Performance Management

Performance management involves measuring the performance of a network and its resources in terms of utilization, throughput, error rates, and response times. With performance management information, a network manager can reduce or prevent network overcrowding and inaccessibility. This helps provide a more consistent level of service to users on the network, without overtaxing the capacity of devices and links.

4. Accounting Management

In the area of accounting management, network monitoring is concerned with gathering usage information to the level of detail required for proper accounting. This type of information helps a network manager allocate the right kind of resources to users, as well as plan for network growth. This type of management also involves monitoring access privileges and usage quotas of the users.

5. Security Management

Security management deals with ensuring the overall security of the network, including protecting sensitive information through the control of access points to that information. Sensitive information is any data that an organization wants to secure, so protecting this sensitive data from unauthorized access is a common requirement.

Security concerns can be assuaged with a well-designed and implemented security management system.[1]

C. STATUS AND FUTURE OF NETWORK MANAGEMENT

The Internet Architecture Board (IAB), which is responsible for networking technology and protocols for the TCP/IP internetworking community, has created a Standard Network Management Protocol (SNMP). This protocol is documented as Request For Comments (RFC's) and is widely published.

The IAB recommends SNMP for use as a common network management protocol with TCP/IP-based networks. Networking with TCP/IP is the most popular type of network and internetworking. As described by RFC 1157, SNMP was initially specified in the late 1980s with three later enhancements that solidified the role of SNMP as the indispensable network management tool.

An enhanced version of SNMP, called SNMPv2 was released in 1993 and revised in 1995. The latest standard called SNMP3, issued in 1998, defines an overall framework for present and future versions by adding security features to SNMP.

The current network management systems are based on the SNMP protocol (Subramanian, 2000). This means that most of the commercial devices that can comprise

[1] SNMP,SNMPv2,SNMPv3, and RMON1 and 2 – William Stalling, Addison-Wesley, 1999 [04]

a computer network support SNMP and they have embedded a SNMP agent. As the agents use TCP/IP, which is a universally accepted protocol and can operate from and through a web-based application, this temporary protocol still remain the main low-level tool for all the management applications.

Nevertheless, it has several limitations that frequently restrict the abilities for monitoring and controlling a network. First of all, it is a polling-based system. This places an extra bandwidth load on the system. Although it seems natural to assume that the more frequently we poll the various stations, the more accurate information we get, this operation is limited by the available control bandwidth on the network. The optimum polling frequency is often the product of a compromise between what we want and what we can obtain.

The need for a NMS is also a limitation. The operating system, manning, and the accessibility over the network restrict the overall efficiency of the system when it is implemented.

An active network, which is the direction of the next generation network, would include embedded network management applications. Fault management and device control applications are of particular importance. As the battlefield network is expanding not only in size but also in complexity a reaction to a single failure must be prompt at least or instantaneous if possible, as the implications can be fast and disastrous. This research concentrates on this particular area, the possible achievement of a self-regulated and self-organized network through the use of AI agents.

D. SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

Simple Network Management Protocol (SNMP) is a distributed-management protocol. A system can operate exclusively as either an NMS or an agent, or the protocol can perform the functions of both. When a system operates as both an NMS and an agent, another NMS might require that system query to manage devices and to provide a summary of the information learned, or that it might require it to report locally stored management information.[2]

[2] SNMP, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm[04]

Since SNMP is the only protocol that manages TCP/IP networks, the following are three other specifications necessary to form the foundation of the management system:

- RFC 1157: A Simple Network Management Protocol that defines the protocol and architecture used to manage the MIB and its contents;
- RFC 1155: Structure and Identification of Management Information for TCP/IP-based networks that describe how the MIB is defined;
- RFC 1213: Management Information Base for Network Management of TCP/IP-based Internets: MIB-II, which describe the contents of the MIB.

This application layer protocol, SNMP, is part of the Transmission Control Protocol/ Internet Protocol (TCP/IP) protocols' suite. It is intended to operate over the User Datagram Protocol (UDP). The management station communicates management information using SNMP, which is implemented on top of the UDP and IP protocols, and a data link protocol, such as Ethernet, Token Ring, or X.25. Likewise, the agent must also implement SNMP, UDP, IP protocols.

The protocol is extensible, allowing developers to add network management functions to their existing projects easily. In addition SNMP separates the management architecture from the architecture of the hardware devices, which broaden the base of multi-vendor support. Unlike other so-called standards, SNMP is not a mere paper specification, but an implementation that is widely available today.[3]

E. THE SNMP ARCHITECTURE

SNMP provides a method of managing network nodes (servers, workstations, routers, bridges, and hubs) from a centrally located host. SNMP performs its management services by using a distributed architecture of management systems and agents. As shown in Figure 1, the centrally located host, which is running network management software, is referred to as an SNMP management system or SNMP manager. Managed network nodes are referred to as SNMP *agents*.

[3] SNMP,http://www2.rad.com/networks/1995/snmp/snmp.htm [04]

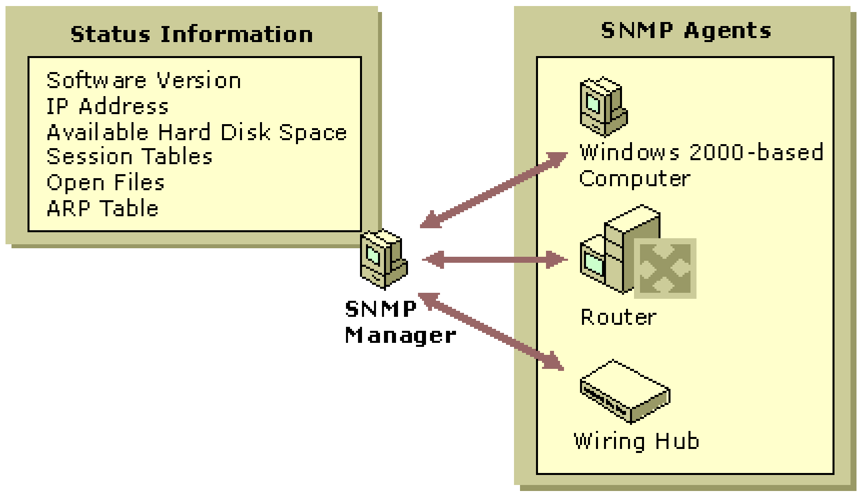


Figure 1. Distributed Architecture of SNMP.[4]

There are four major elements in SNMP network management:

1. Network Management Station

Network Management Station (NMS), typically a stand-alone device, provides the interface for a human network manager to interact with the management system. The management station has the capability to configure, monitor, analyze and control the various components that comprise the network. Some prominent vendors offer network management platforms that implement the role of the network management manager, such as Hewlett-Packard OpenViewTM , IBM NetView, 3 Com Network Supervisor.

2. Management Agent

The management agent is the second active element in the management architecture; it is the workhorse of the SNMP communication. The agent is a software program in the network device that responds to requests for information or actions issued by the management station. The agent may also send the station unsolicited information, known as “Trap.” All devices in a network must have a management agent. Typically, an agent may be embedded or "native" to the device, or alternatively be a "proxy" agent for other protocols.

[4] Figure from http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/tcpip/w2ktcprk.mspx[04]

A proxy agent is the program that supports devices without available SNMP implementation (most mobile devices on the market do not have SNMP implementation). The proxy is an SNMP management that services requests from the management console, on behalf of one or a number of non-SNMP devices.

3. Management Information Base (MIB)

The third part of the architecture is the information exchanged between the manager and the agent; this is called the Management Information Base or MIB. This information is a collection of objects or data values with each representing one aspect of the managed device. For example, the location of the device and the number of erred seconds in the last hour would be two different data values in the MIB. The structure and content of the MIB are standardized across systems of a particular class, such as a bridge MIB or DS-3 MIB. After a MIB is published as a standard, various vendors can build the same kind of equipment that complies with the MIB being assured that they can be managed in a TCP/IP network. The MIB structure is standardized in SNMP as a hierarchical tree. Additions to the tree can be easily accomplished, while traversing a tree to obtain specific information. These are important features because they encourage the use of the MIB in the network management model and the creation of enterprise MIB's for vendors looking to support SNMP with their own products.

The structure of Management Information (SMI), which is given in RFC 1155, is based on the OSI SMI given in Draft proposal 2684. The latest Internet MIB, given in RFC 1213, is called "MIB II." The SMI states that each managed object must have the following elements: a name, syntax and an encoding.

- The name, an object identifier (OID), uniquely identifies the object.
- The syntax defines the data types, such as integer or string of octets. The syntax used for SNMP is the Abstract Syntax Notation One (ASN.1).
- The encoding describes how the information is associated with the managed objects. The encoding used for SNMP is the Basic Encoding Rules (BER).[5]

[5] SNMP, http://www2.rad.com/networks/1995/snmp/snmp.htm [02]

The popularity of SNMP has resulted in the development of standards for storing data critical to network operation, the Management Information Base (MIB). The latest generation of network management MIBs is MIB-II, which stores data on TCP/IP traffic, routing, configuration, and errors. MIB-II has improved support for multi-protocol devices and allows the network management system to control the SNMP operation.

Windows XP Professional includes SNMPv2c extensible agent support, Management Information Base (MIB) II (TCP/IP stack), Host MIB, and a sample MIB. Because SNMP is compatible with Windows NT/2000 source code, MIB's can be ported from the Windows NT/2000 family.

4. Network Management Protocol

The last element of the model, the network management protocol, links the station and the agent by specifying the rules for communication. The protocol used for the management of TCP/IP networks is the SNMP, which uses three simple commands to communicate:

- GET: The basic SNMP request message. Sent by a management system, it requests information about a single MIB entry on an agent — for example, the amount of free drive space.

- GET-NEXT: An extended type of request message that can be used to browse the entire hierarchy of management objects. When it processes a GET-NEXT request for a particular object, the agent returns the identity and value of the object that logically follows the previous information that was sent. The GET-NEXT request is useful mostly for dynamic tables, such as an internal IP route table.

- SET: A message that can be used to send and assign an updated MIB value to the agent when W*rite Access* is permitted.

- GET-BULK: A request that the data transferred by the agent be as large as possible within the given restraints of message size. This minimizes the number of protocol exchanges required to retrieve a large amount of management information.

- NOTIFY: Also called a trap message, NOTIFY is an unsolicited message that is sent by an agent to a management system when the agent detects a certain type of event. For example, a trap message might be sent when a system restart occurs. The management system that receives the trap message is referred to as the trap destination.

Other protocols are available, such as Internet Control Message Protocol (ICMP) and Simple Gateway Monitoring Protocol (SMGP), but they have limited functionality and only support a generic MIB. As a result, these two protocols are not widely used.

F. SUMMARY

The SNMP protocol was designed for the effective, efficient and platform independent monitoring of the network performance. It serves the network administrator by providing the necessary information about critical network performance attributes, variations that can indicate or warn of the network status. The administrator will then take the correct decision that will lead to a stabilized and best configured network.

In the next chapter I explore the capabilities of SNMP regarding the control of the network attributes. Basically I determine whether or not we can change Quality of Service attributes of the network by simply interacting with and changing certain SNMP parameters.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DEVELOPED MODEL DESCRIPTION

A. INTRODUCTION

In order to answer the first question of my research, we developed a model based on the Java programming language. Next we established a small testbed with three PC's based on Microsoft Windows XP Professional Operating system. This operating system supports and provides an SNMP agent based on RFC-1213 MIB. We used one PC as SNMP manager and the rest as managed stations. The results were monitored and they are hereafter cited.

To develop the model, we used a commercial available JAVA API from Advent Inc. The package, AdventNet JAVA API, is totally developed in JAVA, so it can take advantage of the high portability of the Java programming language across various platforms and operating systems. The developed model used the high-level API, which is the API that abstracts from the developer the construction of the UDP datagram. Another approach was used only to verify the result, and this was the Graphic User Interface that is also provided by AdventNet and is called MibBrowser. This GUI is available for evaluation for a limited time, and in this case it was used in the early stages of the experimental procedures.

Before describing the model itself, we describe the implementation of the SNMP protocol within the Microsoft Windows 2000/XP family of operating systems. This is done because of the special way the SNMP is handled within Windows platforms, as there are cases in which attributes are saved within the Windows Registry. In that case, the change or the deletion of these attributes required the execution of a DOS instruction from within the Java model.

B. DESCRIPTION OF THE MODEL

1. Microsoft Windows XP implementation of SNMP protocol

a. Introduction

The Windows XP implementation of SNMP is a 32-bit service that supports computers that are running TCP/IP and IPX protocols. It is an optional service on Microsoft Windows XP Professional and can be installed after TCP/IP and IPX have been successfully configured. Windows XP implements SNMP versions 1 and 2C. These versions are based on industry standards that define how network management information is structured, stored, and communicated between agents and management systems for TCP/IP-based networks.

The Windows XP SNMP service provides an agent that allows centralized, remote management of computers that are running this operating system.

To use the information that Windows XP SNMP service provides, we must have at least one centrally located host that is running an SNMP management software application. The Windows XP SNMP service provides only the SNMP agent; it does not include SNMP management software. We can use some third-party SNMP management software application on the host to act as the management system. Alternatively, we can develop our own SNMP management software application by using the two application programming interfaces (APIs) that are provided with Windows XP:

- WinSNMP API (WinSNMP.dll), which provides a set of functions for encoding, decoding, sending, and receiving SNMP messages.
- Management API (Mgmtapi.dll), which provides a basic set of functions for developing fast and simple SNMP management systems.

The Windows XP SNMP service supports the Internet MIB II, LAN Manager MIB II, Host Resources MIB, and Microsoft proprietary MIBs.

By default, UDP port 161 is used to listen for SNMP messages and port 162 is used to listen for SNMP traps. We can change these port settings by configuring the local Services file. The example illustrated in Figure 2 shows how management systems and agents communicate information.

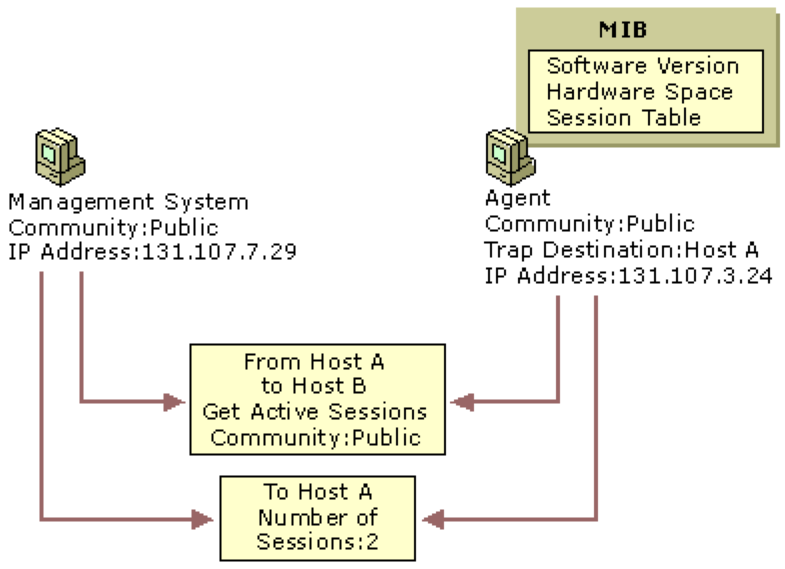


Figure 2. SNMP Manager and Agent Interaction[6]

The communication process is as follows:

- A management system forms an SNMP message that contains an information request (GET), the name of the community to which the management system belongs, and the destination of the message —the agent's IP address (131.107.3.24).
- The SNMP message is sent to the agent.
- The agent receives the packet and decodes it. The community name (Public) is verified as acceptable.
- The SNMP service calls the appropriate subagent to retrieve the session information requested from the MIB.
- The SNMP takes the session information from the subagent and forms a return SNMP message that contains the number of active sessions and the destination — the management system's IP address (131.107.7.29).
- The SNMP message is sent to the management system.

6 Figure from http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/tcpip/w2ktcprk.mspx[04]

b. Communities

Each SNMP management host and agent belongs to an SNMP community. An SNMP community is a collection of hosts grouped together for administrative purposes. Deciding what computers should belong to the same community is generally, but not always, determined by the physical proximity of the computers. Communities are identified by the names one assigns to them.

Community names can be used to authenticate SNMP messages and thus provide a rudimentary security scheme for the SNMP service. Although a host can belong to several communities at the same time, an SNMP agent does not accept requests from a management system in a community that is not on its list of acceptable community names.

There is no relationship between community names and domain names or workgroup names. A community name can be thought of as a password shared by SNMP management consoles and managed computers. It is your responsibility as a system administrator to set hard-to-guess community names when one installs the SNMP service.

Figure 3 illustrates, two communities — Public and Public2. Agent1 can respond to SNMP requests from and can send traps to Manager2 because they are both members of the Public2 community. Agent2, Agent3, and Agent4 can respond to SNMP requests from and can send traps to Manager1 because they are all members of the (default) Public community.

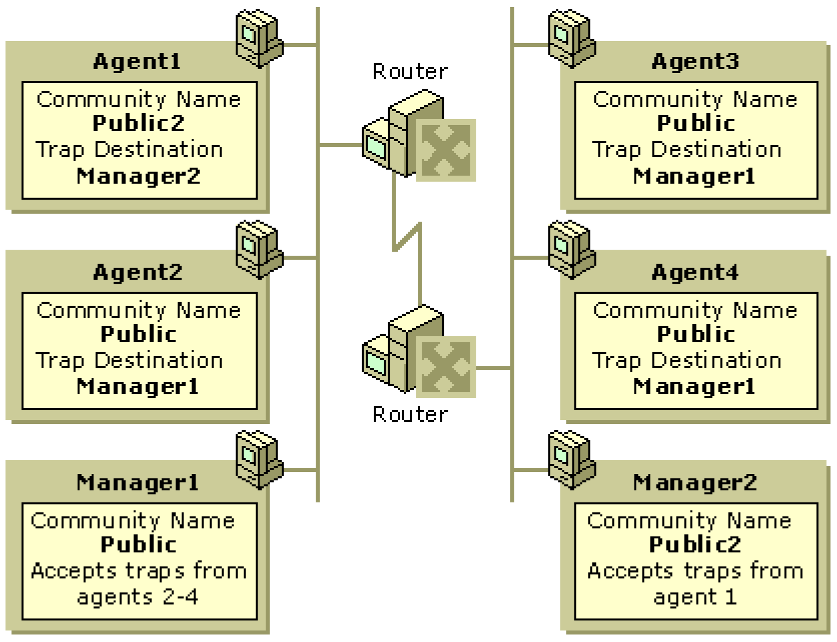


Figure 3. Example of SNMP Communities[7]

Community names are managed by configuring the SNMP security properties.

When an SNMP agent receives a message, the community name contained in the packet is verified against the agent's list of acceptable community names. After the name is determined to be acceptable, the request is evaluated against the agent's list of access permissions for that community. The types of permissions that can be granted to a community include the following:

- None: The SNMP agent does not process the request. When the agent receives an SNMP message from a management system in this community, it discards the request and generates an authentication trap.
- Notify: This is currently identical to the permission of None.
- Read Only: The agent does not process SET requests from this community. It processes only GET, GET-NEXT, and GET-BULK requests. The agent

[7] Figure from http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/tcpip/w2ktcprk.mspx[04]

discards SET requests from manager systems in this community and generates an authentication trap.

- Read Create: The SNMP agent processes or creates all requests from this community. It processes SET, GET, GET-NEXT, and GET-BULK requests, including SET requests that require the addition of a new object to a MIB table.

- Read Write: Currently identical to Read Create.

c. Architecture of Windows XP SNMP

The internal architecture of the Windows XP implementation of SNMP is divided into management and agent functions; in some cases, these functions overlap, as illustrated in Figure 4.

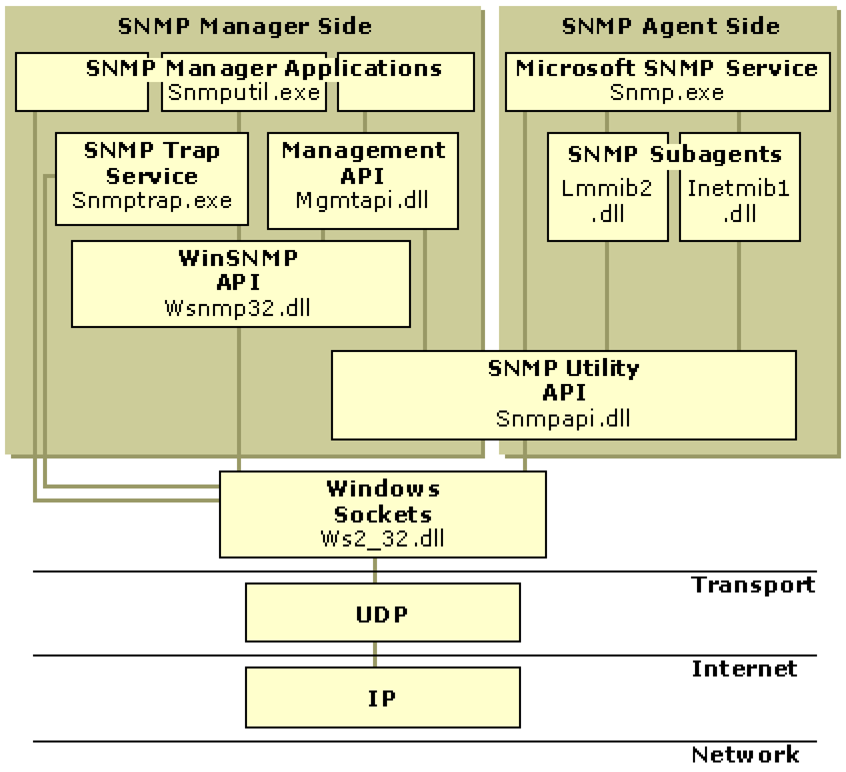


Figure 4. Windows 2000 SNMP Architecture[8]

[8] Figure from http://www.microsoft.com/resources/documentation/windows/2000/server/reskit/en-us/tcpip/w2ktcprk.mspx[04]

d. Registry Settings

The SNMP service converts the information in the registry into a format that can be used by third-party SNMP network management programs. Whenever possible, use the Windows XP SNMP service user interface to alter service settings. When changes are made to SNMP service properties through the user interface, the corresponding SNMP registry settings are modified, with the exception of the following registry setting, which defines the list of extension agents (subagents) that are configured:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SNMP\Parameters
\ExtensionAgents
```

The SNMP service detects any registry changes while running. The SNMP parameter changes are activated without the need to restart the SNMP service.

2. Creating SNMP Manager Using AdventNet Java Libraries

a. Introduction to AdventNet Java API

AdventNet SNMP API is a comprehensive toolkit for rapid development of SNMP-based management applications that are reliable, scalable, and independent of operating system.

Network management developers can leverage AdventNet SNMP library to build standalone and Web-based applications, embedded software components and distributed EJB, CORBA, and RMI applications. The library provides many of the commonly used functions and components out-of-the-box to make the development simpler.

The core of AdventNet SNMP API is a set of Java APIs that can be integrated in any typical Java application. In addition, the APIs provide interfaces that enable deployment in distributed environment through RMI, CORBA, and J2EE containers. Built using the best software design patterns and optimized performance, it is a powerful suite to secure APIs to build cross-platform, real-time application for monitoring and tracking the performance of network elements.

b. About AdventNet API

AdventNet SNMP API can be used for building management applications for delivering solutions that are appropriate for specific needs in the spheres of Internet

Infrastructure Management. This API can be used and integrated into any programming infrastructure that might require network management solution.

The benefits of using the AdventNet API are

- Cross-platform Support: supports all major platforms, such as Solaris, Windows NT, and Linux with a common code base.

- Open Standards: follows the current Internet and standard technologies, such as Java beans, HTTP, RMI, CORBA, EJB, and JDBC. The key benefit of these technologies is the feature-rich, easy-to-use, and developer-friendly API resulting in faster time to market and easier development of custom network management solutions.

- Fast Track Development: simplifies the creation and deployment of SNMP management applications using the SNMP Design Studio, an easy-to-use visual development environment. Auto code generation reduces the scope for human errors in the source code thereby cutting time and cost on development, debugging, and testing. SNMP Design Studio also provides in-built tools for code editing, debugging, testing, maintenance, and packaging.

- Customization: provides a rich set of Java interfaces to deliver a highly customizable solution.

- Scalability: fundamentally designed as a multi-tier system, based on widely used Internet technologies for highly scalable solutions. The latest release of AdventNet SNMP API provides EJB support thus enabling building of distributed applications.

- Flexibility: provides a hierarchy of Java library packages, which allow flexible selection of the level of library support desired. Therefore, you can access the detailed SNMP information by using low-level API, or choose higher-level Java Beans for simpler programming and additional functionality which requires no dealing with the SNMP details.

- Web-based Network Management: includes modules, such as SAS and HTTP which allow one to manage the network on the Internet or even to manage devices behind firewalls. The SAS APIs can also be extended to add SSL support for secure management.

- Conformance to Standards: AdventNet SNMP API conforms to most Internet RFC specifications.

Finally, the MibBrowser tool can be used both as an application and as an applet.

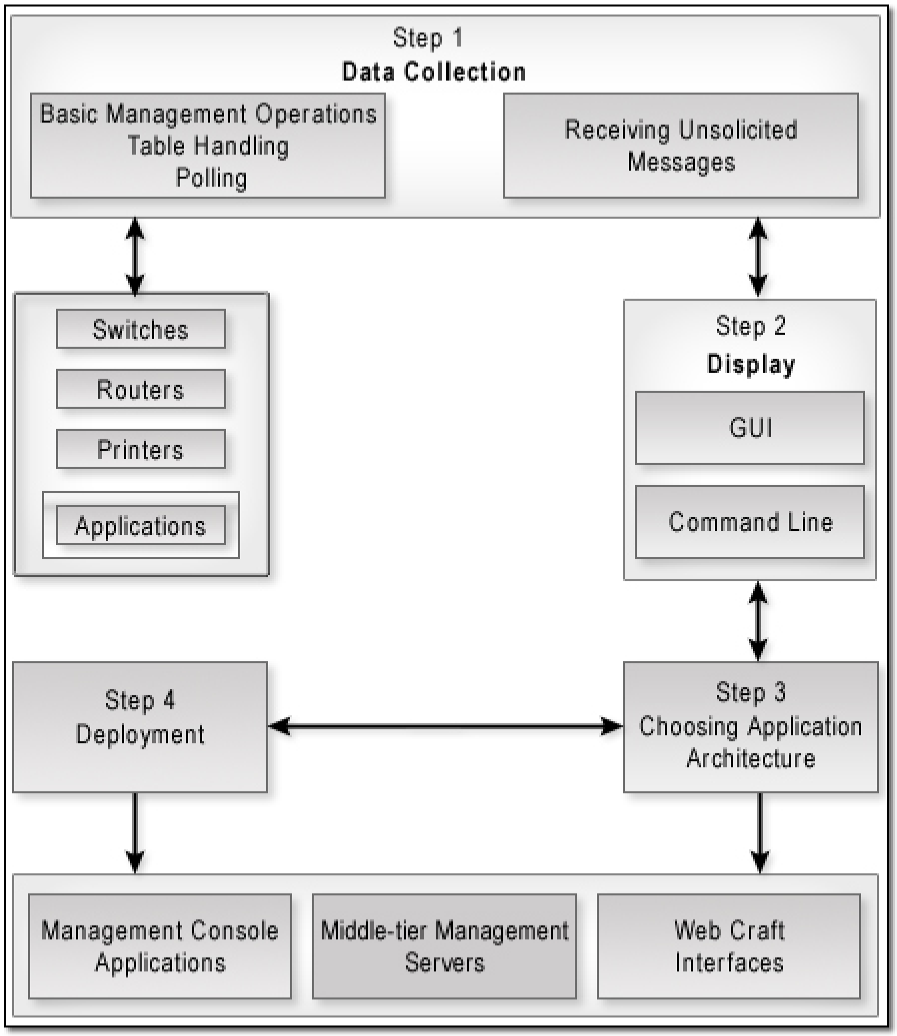

Step 1
Data Collection
Basic Management Operations
Table Handling
Polling
Receiving Unsolicited Messages
Switches
Routers
Printers
Applications
Step 2
Display
GUI
Command Line
Step 4
Deployment
Step 3
Choosing Application Architecture
Management Console Applications
Middle-tier Management Servers
Web Craft Interfaces

Figure 5. Architecture of the Management Application [9]

[9] Figure from AdventNet help files[04]

c. AdventNet SNMP API Architecture

SNMP API can be used with two-tier as well as three-tier management applications. In the two-tier architecture, the management applications directly communicate with the agents. In the three-tier architecture, the management applications communicate with the agents through a manager-server. For building highly scalable management applications, three-tier architecture is the best option.

AdventNet SNMP API provides a wide choice of options in selecting the type of architecture the management application needs. The Choosing the Application Architecture section discusses the various types of network management applications and applets that can be developed using AdventNet SNMP API.

(1) Deployment Options. In today's distributed environment, applications should have a wide choice of deployment options. Deployment of standalone applications is preferred for most environments, while applet deployment might be needed for Web-based management of network entities.

Management applications developed using AdventNet SNMP API can be deployed in different formats, such as applications, applets, and servers. Support for applets in AdventNet SNMP API is provided by means of SNMP Applet Server (SAS). SAS enables communication between applets on Java browsers, which do not permit socket access to any host other than the applet host.

(2) Accessing the Data from the MIBs Supported by the Agent. The management applications typically request management data or properties from one or more SNMP-enabled nodes. The application needs to know the names and types of objects in the managed device. This is available in the Management Information Base (MIB) modules, which are usually provided with the managed devices.

The data supported by the agent are available in the form of MIB files and the manager applications make use of the information available in the MIB files while querying the agent. For example, RFC1213-MIB, also known as MIB-II, is a MIB module that is supported by all SNMP agents on TCP/IP-enabled devices. Apart from supporting MIB-II, each device has its own MIB, such as a printer MIB, a modem MIB, or a switch MIB. These MIBs can be used to access the associated data with it.

- Management applications should be able to:
- Load and unload MIB modules
- Access the information on managed objects using MIB
- Resolve the textual labels to numerical OIDs
- Determine the type of data of the MIB object

Simple management applications normally make a request by manually loading the MIB file, entering the OID, data type, and data value of the each variable binding. Advanced applications require loading multiple MIBs, storing the MIBs, logging the management requests, and so on.

AdventNet SNMP API provides rich support for handling and manipulating MIBs. The MIB support package of AdventNet SNMP API is designed to allow Java programs to take full advantage of the information contained in the MIB files. The Using MIBs in Applications chapter discusses more on MIB-related aspects while developing the SNMP management applications.

(3) Collecting Data from the Agent. The SNMP management applications communicate with the agents to retrieve the data. Applications normally retrieve the data by synchronous/asynchronous communication or by polling at regular intervals.

The applications, while communicating through the synchronous mode, wait for the response of the previous request before sending a new request. In the case of asynchronous mode, the manager application can keep sending requests to the agent without waiting for the response. The responses are retrieved using the callback mechanism.

Though using asynchronous mode appears to lead to improved performance, it can be used when the manager application knows the OID of the object it has to query. The synchronous operation is relevant while retrieving something like a tabular data, in which the instance of the OID to be queried next is obtained from the response of the previous request.

AdventNet SNMP API has a comprehensive hierarchy of Java packages that allows a flexible selection of the desired level of library support. The developer can access the detailed SNMP information or choose higher-level Java Beans for simpler programming.

In the case of low-level APIs, the user has to handle all the resources including the low-level resources, such as session, api, pdu, and miboperations. The disadvantage of this is the complexity of the code to be written for developing a management application.

While using high-level API, the developers need not handle the low-level resources and they can use the API methods of the beans package for all operations.

API Overview explains the architecture of the different modules available as part of the AdventNet SNMP API distribution and their functions and features.

The data retrieving functions of the manager applications can be classified as follows:

- Communicating with SNMP agent
- Table handling
- Polling

(4) Communicating with SNMP Agent. Management applications normally retrieve the properties of the devices using SNMP GET, GETNEXT, or GETBULK request to the OIDs. The request may be simply to check whether the node is alive or to retrieve the values of specific managed objects periodically.

The sections Data Retrieval Operations and Data Altering Operations discuss how the applications communicate with the agent to access the data and the various ways of accessing the data using the SNMP protocol.

(5) Table Handling Most of the MIBs are designed to handle large data in the form of tables. Management applications should be able to retrieve tables quickly and efficiently. Intuitive table handling GUIs should become part of the management applications.

Table Handling in Applications explains the various table-related operations that can be performed using AdventNet SNMP API.

(6) Polling. The retrieval of data for specific managed objects at periodic interval of time is called polling. Polling is normally used to monitor data that may change over time. Repeated polling of data is required when the object is a critical resource or when it is required to monitor the performance.

Polling can be performed for one or more nodes. Polling is started by selecting the specific node, the OID to be polled, the timeout-retry values, and the polling interval. The polling interval is normally given in seconds or in minutes. The polling interval can be determined by the size and the number of messages sent to each polling cycle, the time taken by the agent to respond, and so on. The application can be used to poll the agent regularly, watch for threshold crossings, and to take an appropriate action based on the results.

Data Collection and Reporting discusses the polling features that are available in AdventNet SNMP API.

(7) Displaying the Retrieved Data. After the necessary data is collected from the agent, the management application has the option of displaying the result in the form of UI or non-UI. In case of displaying the results in the form of UI, the users have to build their own UI components.

AdventNet SNMP API comes with built-in UI components, such as LineGraph, BarGraph, TrapViewer, SnmpTablePanel, and MibBrowser to display the data received from the device.

For displaying the results in non-UI format, the developers can choose high-level non-UI beans or directly use the low-level API.

The advantage of using the high-level API is that the bean components perform the most common functions, such as splitting the PDU in the case of large requests/responses, removing the error OIDs from multi-varbind requests, returning the results for proper varbinds, retrieving table data, receiving traps at specified ports, and so on.

d. AdventNet Architecture

AdventNet SNMP API is the most comprehensive development environment for building SNMP-based management applications and applets. The API consists of a hierarchy of Java packages that can be used for developing Java-based and Web-based network management products and solutions.

The following image illustrates the organization of the SNMP API architecture:

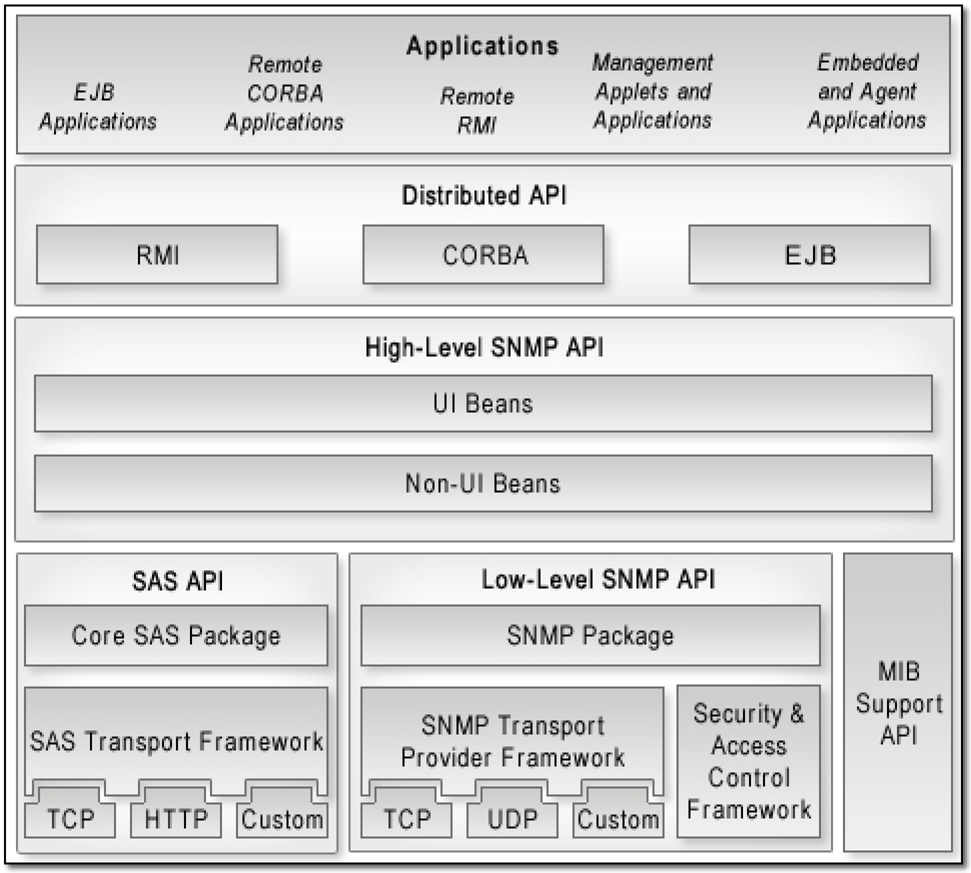


Figure 6. High Level API [10]

(1) High-Level API. The high-level API consists of UI and non-UI beans that can be used to build applications and applets that incorporate the SNMP functions provided by the low-level API. These bean components can be used in any Java Bean Builder or directly in the Java code and can be used in developing management applications. The components are built using the functions provided by the low-level API and MIBs API.

The UI beans can be used in developing management GUI applications. The beans provided in this package have SNMP intelligence. This allows one to build more flexible applications, applets, and components. The non-UI beans form the backbone of the high-level API and the UI beans are built on top of the non-UI beans.

[10] Figure from AdventNet help files[04]

(2) Low-Level API. The low-level API implements the core functions of the protocol. It includes classes that facilitate communication with peer SNMP entities and offer message security and privacy to applications and applets. It also includes classes that can be used in management applets running in a browser. It supports multilingual communication with devices.

The low-level API provides the reference implementation of USM and VACM for SNMPv3 entities. It also offers protocol-independent communication framework for SAS communication, in which you can plug in your transport protocol for SAS communication.

(3) MIBs API. The MIBs API conveys the information about the data available on an SNMP agent. This API allows Java programs to take full advantage of the information contained in MIB module files. It also facilitates loading and unloading MIBs in applications and applets, in addition to supporting a host of functions that provide the properties of the managed object. The components are built using the variable support functions provided by the low-level API.

C. STRUCTURE OF THE EXPERIMENT

1. Experiment Procedure

- Enable the SNMP service on PC clients both Windows 2000 and Windows XP Professional.
- Test whether SNMP agent on clients can communicate with the SNMP manager.
- Load RFC1213-MIB II on SNMP manager.
- Test whether the SNMP manager can retrieve SNMP information on SNMP agents.
- Test whether by altering specific SNMP attributes, we can change the network topology and/or the network QoS attributes (bandwidth allocation and/or routing tables).

2. Hypothesis

Changing the network QoS attributes by directly altering the information of MIB’s is feasible. If so, then we can achieve a direct two-way communication with SNMP agents and by processing the data that they provide through polling, we can alter specific the SNMP attributes in order to achieve an adaptive situation in the network topology.

D. IMPLEMENTATION

First we established a small network with three PC’s and a manager station. The operating system of choice was Windows XP Professional, as there was a need for the latest implementation of Microsoft SNMP Agent. There was no need for the presence of a PC acting as a server because the SNMP managing operation could be well done by any of the four PC’s.

Actually we could have only two PC’s to establish the necessary network for our experiment but then we could not exploit the TRAP command.

After we established the Ethernet topology for the four PC’s using a router, we proceeded by activating the SNMP agent on all the machines, following the procedures described in Annex 1.

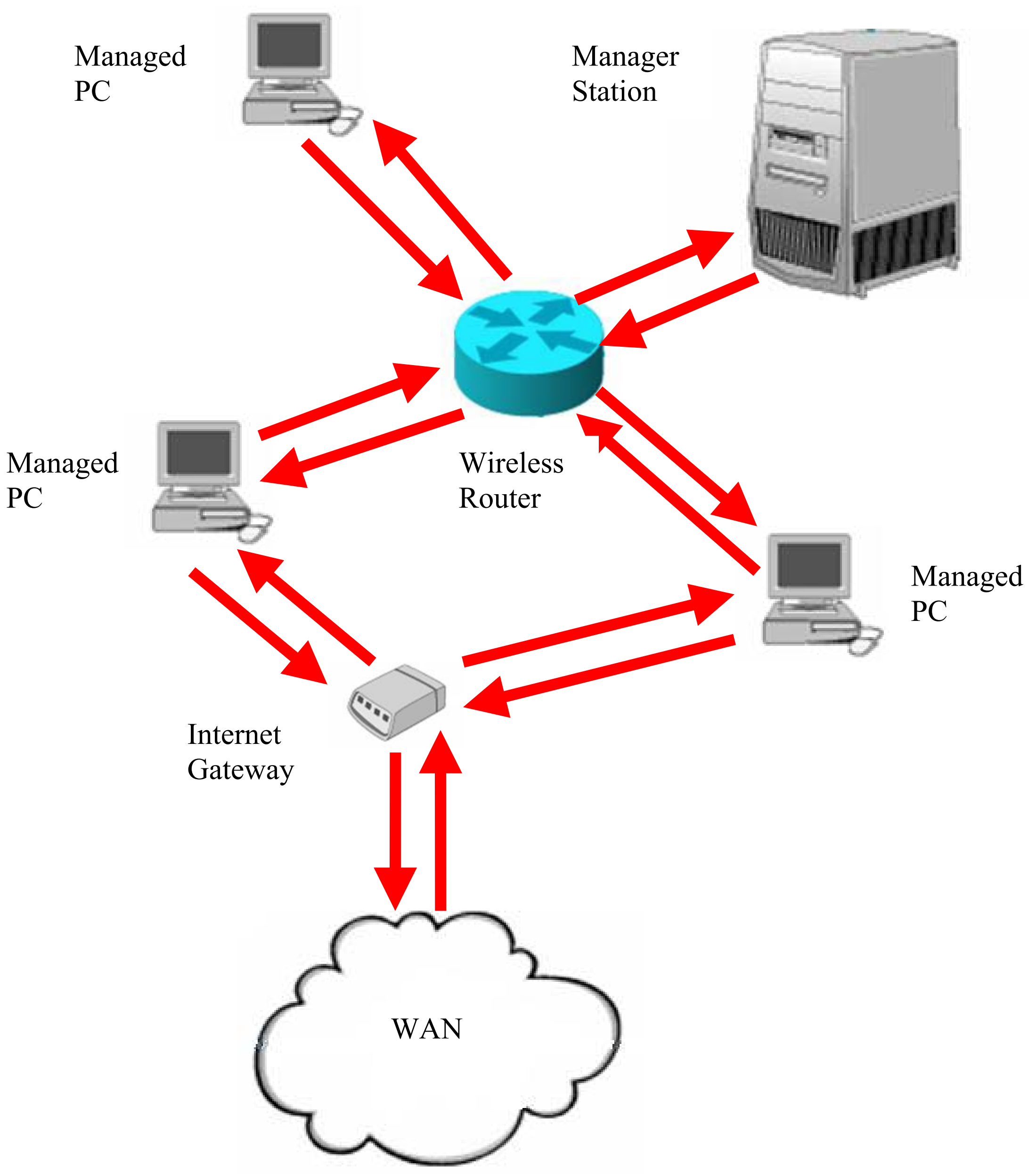


Figure 7. Experiment Topology Diagram

Then we used one PC as a manager and we initialized the MibBrowser application. AdventNet MibBrowser is an interface for accessing MIB attributes of a specific client. As the Microsoft SNMP agent implements SNMPv1 and SNMPv2c, we can easily access SNMP data of a client by polling an attribute of the loaded MIB. The SNMP agent is activated by default at the local port 161 as shown at Figure 7. SNMP object ids, types, and values can be retrieved either by using arrays of numbers or arrays of descriptive names. For example, the value for the System Description can be retrieved either as 1.3.6.1.2.1.1.1.0 or .iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0.

The Java code that was used for the experiment can be found at the Annex 2.

E. EXPERIMENTAL TESTING

1. Enable SNMP Service

We first enabled the SNMP service. The procedure is cited at Annex 1. We also created a new READ/WRITE community for the purpose of testing in order to be able to access the MIB and change the attributes we want. We named the READ/WRITE community as "harman." We repeated the same procedure on all three PC's of the network topology. Now we could access the SNMP agent of any of the three clients from any other PC within the network. If we had a PC within another LAN or a WAN with the same community name, then the PC could be managed by using the community name and the IP address.

2. Test Manager – Agent Communication

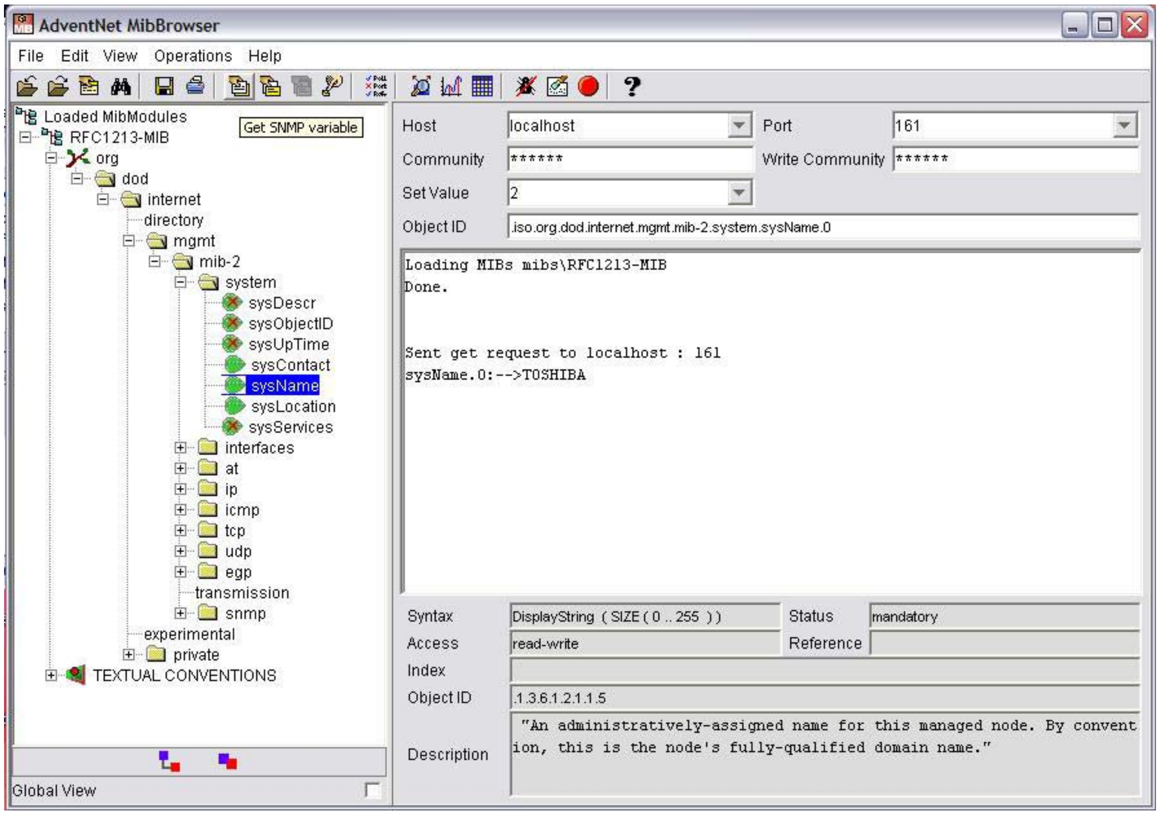
Then we issued specific polls to all three PC's from the manager. The polls regarded the name of each PC and the respond was recorded as successful because all clients, even the loopback, responded correctly

3. Load RFC1213-MIB

The initialization of the MibBrowser comes with the default loading of RFC1213-MIB known also as MIB-II. This specific MIB is supported by the SNMP agent that is available with the Microsoft Windows XP/2000 and 2003 Server operating systems.

4. SNMP Attributes Polling Testing

We can use the tree-like structure of the IDE to walk through the tree of the specific MIB. After clicking on a specific leaf, we can get the SNMP variable related to a specific host and community by polling the SNMP agent through the IDE. As a first step we try to access the information related to the System Name of a specific client. As we can see in Figure 8, the IDE gives us the answer "TOSHIBA," which is correct in the case of the local host loopback.



Host	localhost	Port	161
Community	******	Write Community	******
Set Value	2		
Object ID	.iso.org.dod.internet.mgmt.mib-2.system.sysName.0		

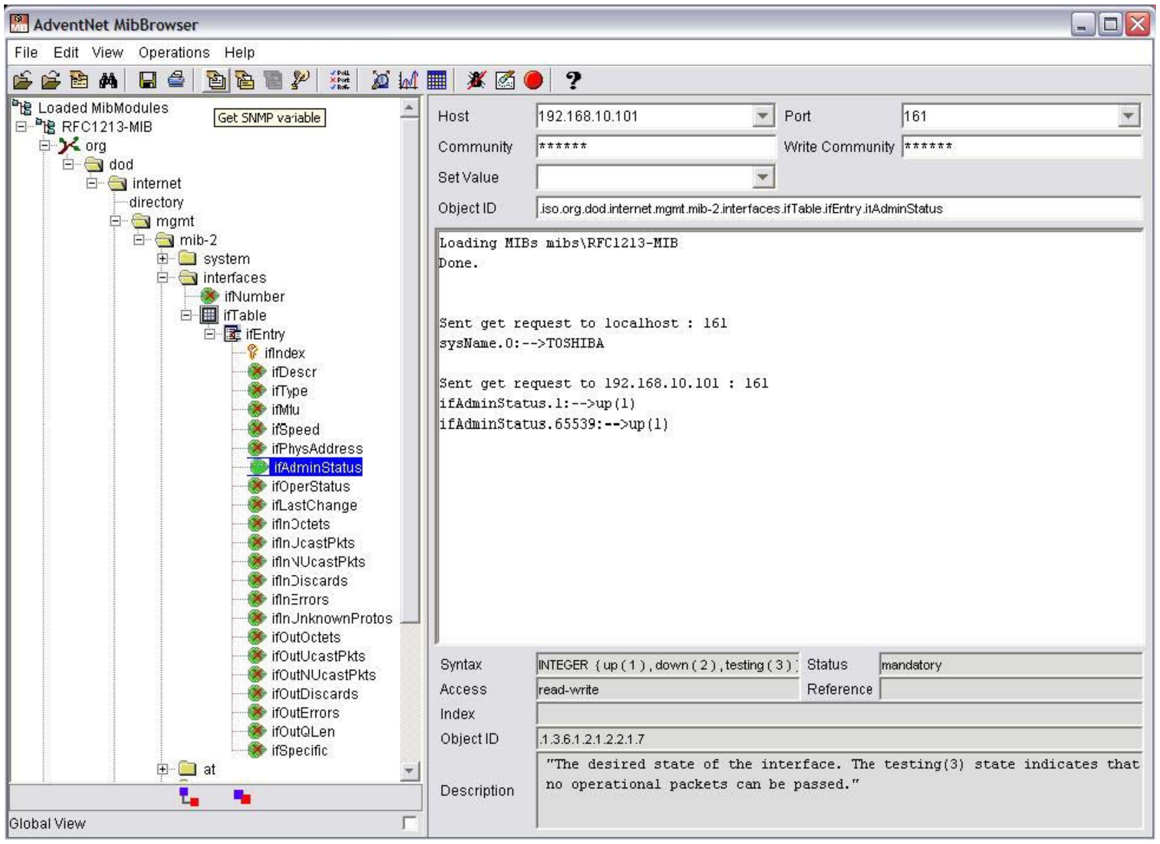
```
Loading MIBs mibs\RFC1213-MIB
Done.


Sent get request to localhost : 161
sysName.0:-->TOSHIBA
```

Syntax	DisplayString (SIZE (0 .. 255))	Status	mandatory
Access	read-write	Reference	
Index			
Object ID	.1.3.6.1.2.1.1.5		
Description	"An administratively-assigned name for this managed node. By convention, this is the node's fully-qualified domain name."		

Figure 8. MibBrowser Initialization

We can already see from the initialization screenshot, the leaves of a specific MIB tree branch. These leaves have or do not have an additional red X which means that this specific attribute is READ ONLY or READ-WRITE respectively. This means that even if we establish a write community for a specific client and we access this client with the intention of altering a specific attribute, and then if this attribute is READ ONLY, it can not be changed.

Figure 9 shows SNMP data related to the IP layer of the TCP/IP stack. There we can see that although we can obtain information about each connection, we have very limited capabilities of intervening. This means that we may get what the connection from and to the specific client are, what the payload of each connection is, how many missed or bad packets each connection received, but the only attribute we can change is whether this connection will be up or down. In other words, we cannot change the rate of down/uploading neither the consumed bandwidth.



Host 192.168.10.101 Port 161
Community ****** Write Community ******
Set Value
Object ID .iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus

```
Loading MIBs mibs\RFC1213-MIB
Done.


Sent get request to localhost : 161
sysName.0:-->TOSHIBA

Sent get request to 192.168.10.101 : 161
ifAdminStatus.1:-->up(1)
ifAdminStatus.65539:-->up(1)
```

Syntax INTEGER { up (1) , down (2) , testing (3) Status mandatory
Access read-write Reference
Index
Object ID .1.3.6.1.2.1.2.2.1.7
Description "The desired state of the interface. The testing(3) state indicates that no operational packets can be passed."

Figure 9. SNMP Attribute GET Operation

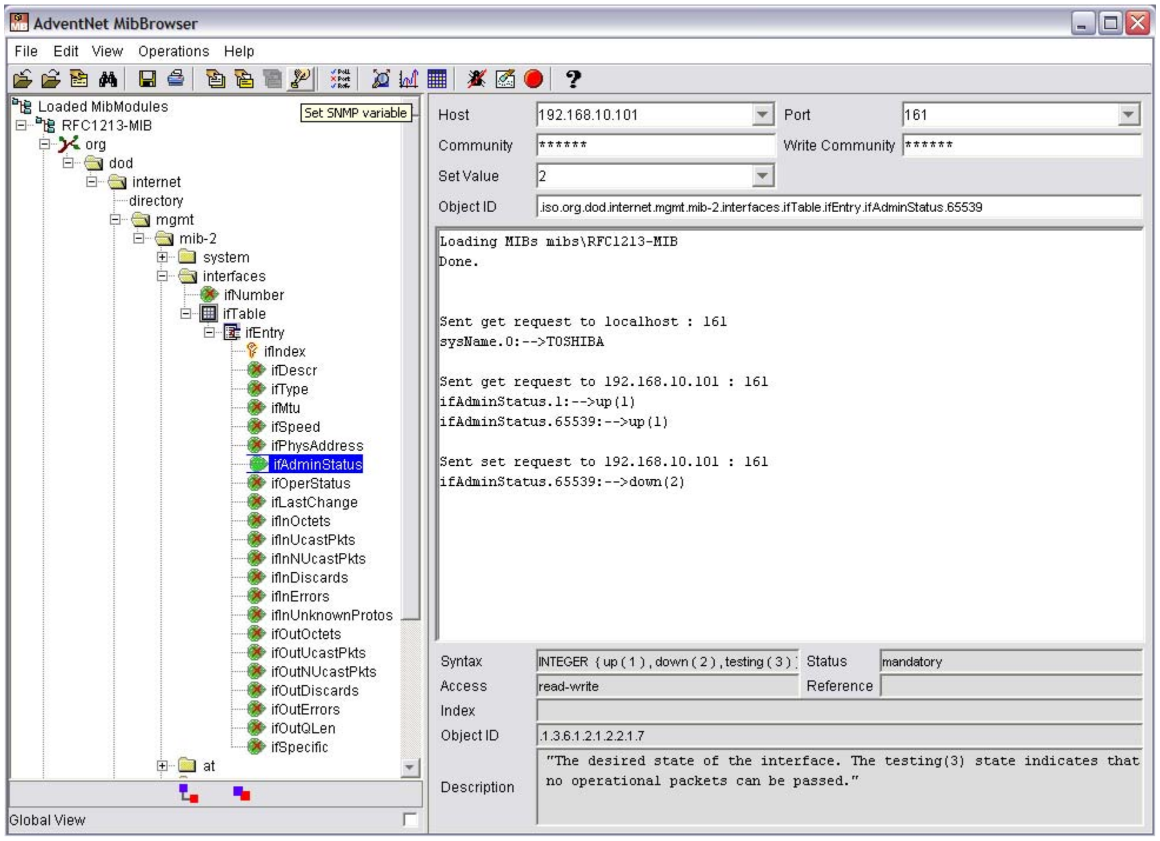
5. SNMP Attributes Set Testing

Next we will try to change the specific values of the SNMP attributes. We will use the SET instruction or even better the IDE of the MibBrowser. In this case we have to establish first a READ/WRITE community and correctly input its name in the proper window.

At first we will try to change the value of the polled attribute from the previous testing. Therefore we set the value of:

.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus.65538

at 2, which stands as DOWN for the specific interface. The SET operation is shown in Figure 10 and the results in Figure 11.



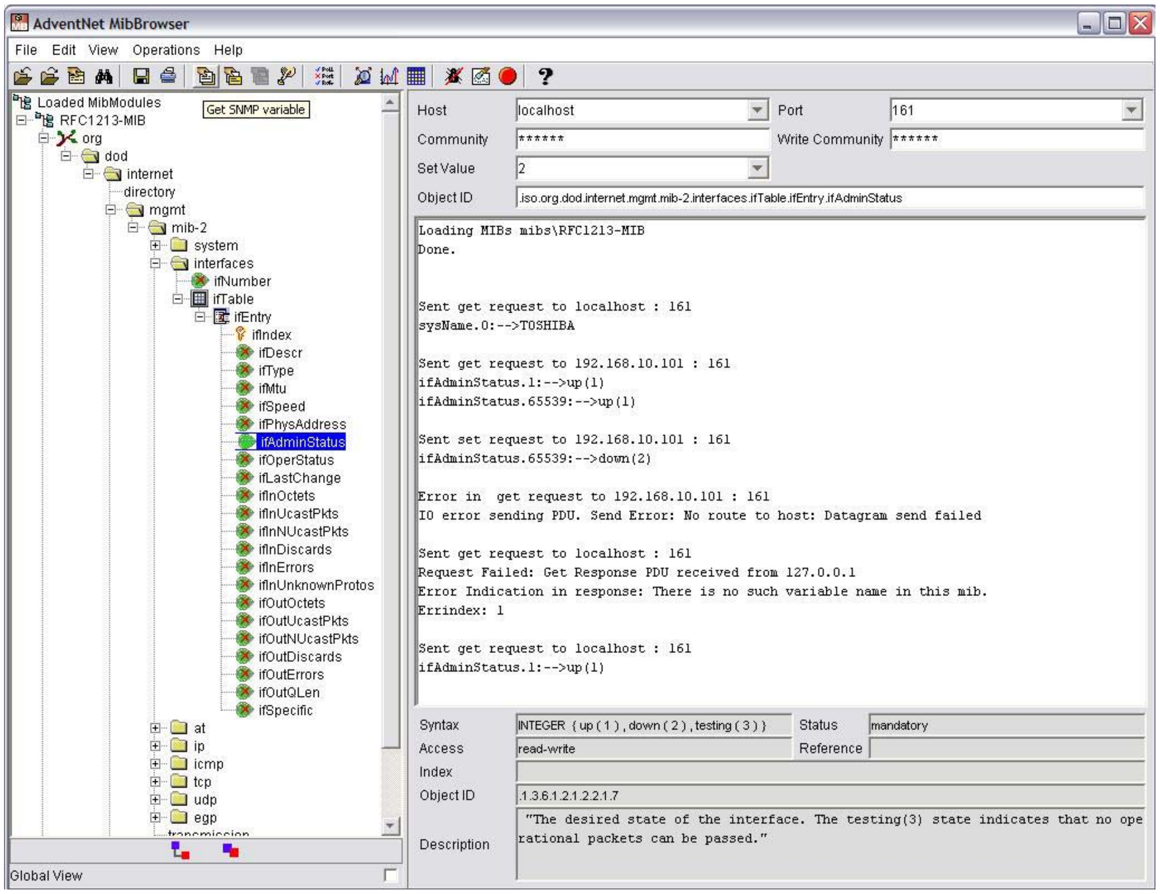
Host: 192.168.10.101
Port: 161
Community: ******
Write Community: ******
Set Value: 2
Object ID: .iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus.65539

```
Loading MIBs mibs\RFC1213-MIB
Done.


Sent get request to localhost : 161
sysName.0:-->TOSHIBA

Sent get request to 192.168.10.101 : 161
ifAdminStatus.1:-->up(1)
ifAdminStatus.65539:-->up(1)

Sent set request to 192.168.10.101 : 161
ifAdminStatus.65539:-->down(2)
```

Syntax: INTEGER { up (1) , down (2) , testing (3)
Status: mandatory
Access: read-write
Reference:
Index:
Object ID: .1.3.6.1.2.1.2.2.1.7
Description: "The desired state of the interface. The testing(3) state indicates that no operational packets can be passed."

Figure 10. SNMP Attribute SET Operation

We can observe that when we tried to access information about the status of the Wireless Interface, there was a time-out response because this interface no longer existed. Then we tried to access information using the loopback address, and we saw that actually there was only one IP interface left at the client. This happened although there was an icon on the Windows Taskbar.



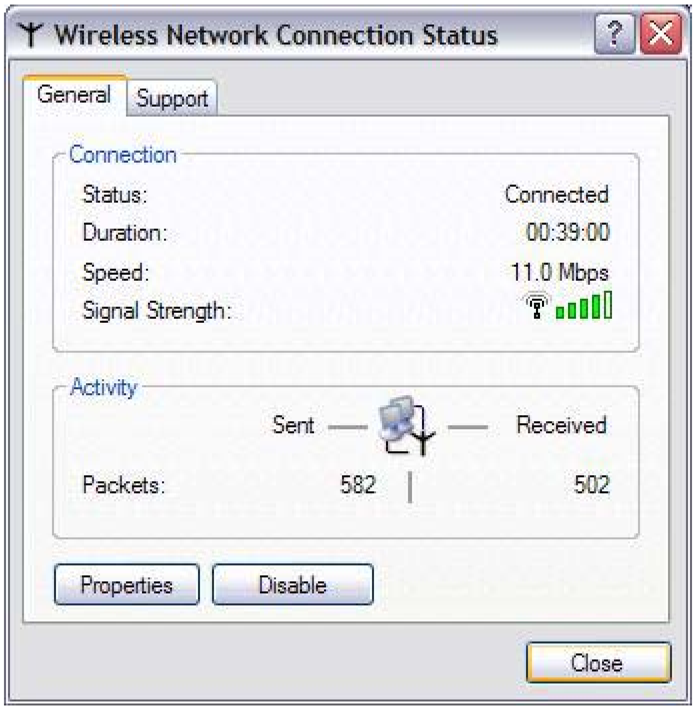
Host	localhost	Port	161
Community	******	Write Community	******
Set Value	2		
Object ID	.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifAdminStatus		

```
Loading MIBs mibs\RFC1213-MIB
Done.


Sent get request to localhost : 161
sysName.0:-->TOSHIBA

Sent get request to 192.168.10.101 : 161
ifAdminStatus.1:-->up(1)
ifAdminStatus.65539:-->up(1)

Sent set request to 192.168.10.101 : 161
ifAdminStatus.65539:-->down(2)

Error in  get request to 192.168.10.101 : 161
IO error sending PDU. Send Error: No route to host: Datagram send failed

Sent get request to localhost : 161
Request Failed: Get Response PDU received from 127.0.0.1
Error Indication in response: There is no such variable name in this mib.
Errindex: 1

Sent get request to localhost : 161
ifAdminStatus.1:-->up(1)
```

Syntax	INTEGER { up (1) , down (2) , testing (3) }	Status	mandatory
Access	read-write	Reference	
Index			
Object ID	.1.3.6.1.2.1.2.2.1.7		
Description	"The desired state of the interface. The testing(3) state indicates that no operational packets can be passed."		

Figure 11. SNMP Attribute SET Operation Results

Nevertheless, when we tried to access the properties window of the connection, we found that the General tab showed that everything was normal and indeed there was a connection. But the Support tab indicated that there was no connection at all and so there was no IP address assigned. The two tabs, General and Support, of the Wireless Network Connection Status are shown in Figure 12 and Figure 13 respectively.



Connection

Status: Connected

Duration: 00:39:00

Speed: 11.0 Mbps

Signal Strength:

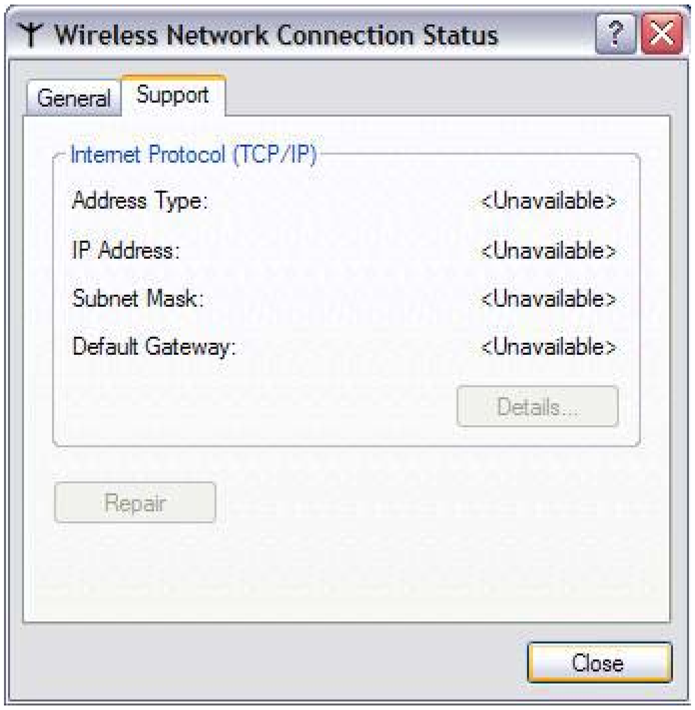
Activity

Sent — — Received

Packets: 582 | 502

Figure 12. Wireless Network Connection Status: General Tab

The two different views of the same connection can be explained by the fact that we intervened at the IF portion of the TCP/IP stack and so the Windows retained the TCP portion without any change. Although the connection was lost and we could not access any other client or server through this connection, this behavior must a be Windows specific implementation issue of the TCP/IP stack.


Wireless Network Connection Status
General
Support
Internet Protocol (TCP/IP)
Details...
Repair
Close

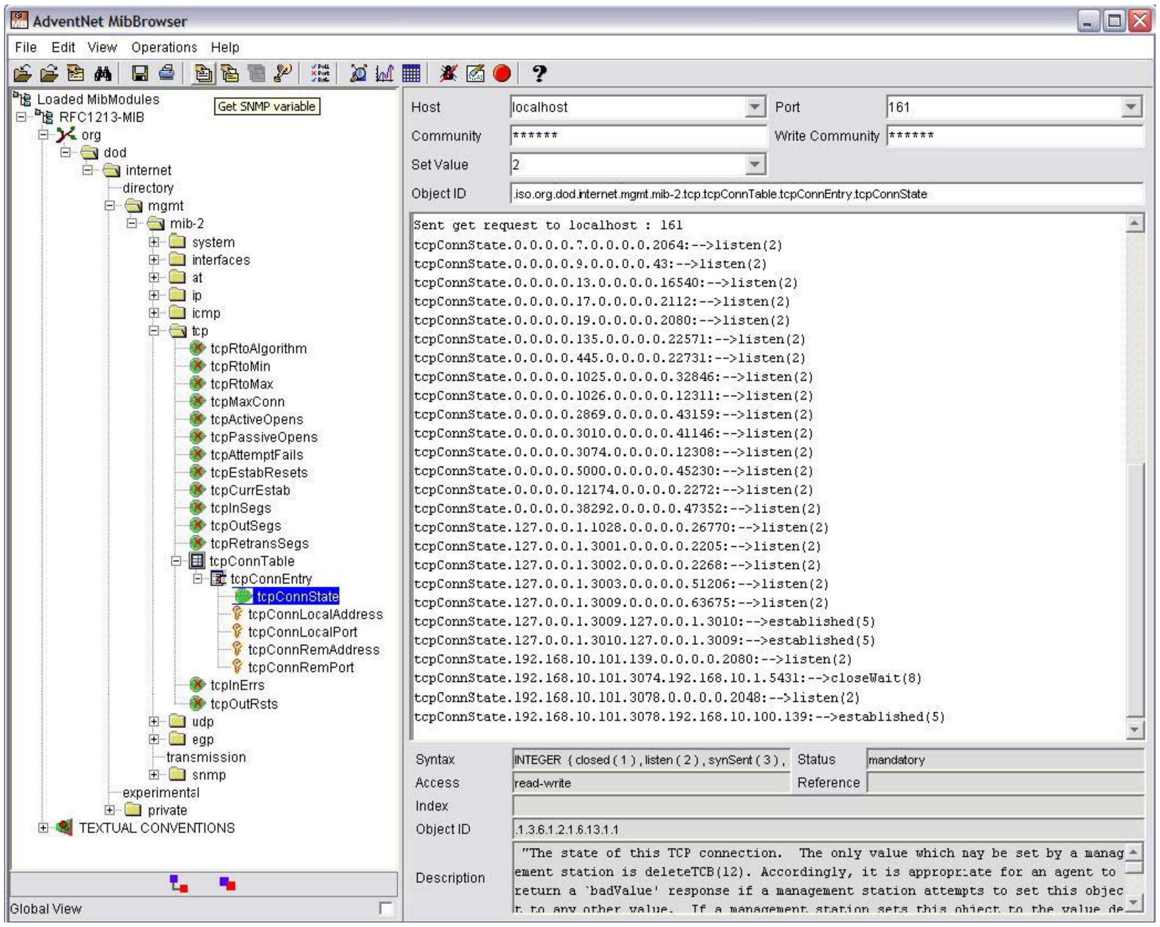
Address Type: <Unavailable>
IP Address: <Unavailable>
Subnet Mask: <Unavailable>
Default Gateway: <Unavailable>

Figure 13. Wireless Network Connection Status: Support Tab

This was the only attribute that we could change from the IF sub-tree of the RFC1213-MIB. We also tried to change the values at the TCP level but the only WRITE attribute is

.iso.org.dod.internet.mgmt.mib-2.tcp.tcpConnTable.tcpConnEntry.tcpConnState

The only option available about this attribute is to close a specific connection. There is no possibility of opening a specific connection with a source/destination port and a source/destination IP address. Although we tried to close a specific connection, the results proved that it could not be achieved. This can be seen in Figure 14.



Host: localhost
Port: 161
Community: ******
Write Community: ******
Set Value: 2
Object ID: .iso.org.dod.internet.mgmt.mib-2.tcp.tcpConnTable.tcpConnEntry.tcpConnState

```
Sent get request to localhost : 161
tcpConnState.0.0.0.0.7.0.0.0.0.2064:-->listen(2)
tcpConnState.0.0.0.0.9.0.0.0.0.43:-->listen(2)
tcpConnState.0.0.0.0.13.0.0.0.0.16540:-->listen(2)
tcpConnState.0.0.0.0.17.0.0.0.0.2112:-->listen(2)
tcpConnState.0.0.0.0.19.0.0.0.0.2080:-->listen(2)
tcpConnState.0.0.0.0.135.0.0.0.0.22571:-->listen(2)
tcpConnState.0.0.0.0.445.0.0.0.0.22731:-->listen(2)
tcpConnState.0.0.0.0.1025.0.0.0.0.32846:-->listen(2)
tcpConnState.0.0.0.0.1026.0.0.0.0.12311:-->listen(2)
tcpConnState.0.0.0.0.2869.0.0.0.0.43159:-->listen(2)
tcpConnState.0.0.0.0.3010.0.0.0.0.41146:-->listen(2)
tcpConnState.0.0.0.0.3074.0.0.0.0.12308:-->listen(2)
tcpConnState.0.0.0.0.5000.0.0.0.0.45230:-->listen(2)
tcpConnState.0.0.0.0.12174.0.0.0.0.2272:-->listen(2)
tcpConnState.0.0.0.0.38292.0.0.0.0.47352:-->listen(2)
tcpConnState.127.0.0.1.1028.0.0.0.0.26770:-->listen(2)
tcpConnState.127.0.0.1.3001.0.0.0.0.2205:-->listen(2)
tcpConnState.127.0.0.1.3002.0.0.0.0.2268:-->listen(2)
tcpConnState.127.0.0.1.3003.0.0.0.0.51206:-->listen(2)
tcpConnState.127.0.0.1.3009.0.0.0.0.63675:-->listen(2)
tcpConnState.127.0.0.1.3009.127.0.0.1.3010:-->established(5)
tcpConnState.127.0.0.1.3010.127.0.0.1.3009:-->established(5)
tcpConnState.192.168.10.101.139.0.0.0.0.2080:-->listen(2)
tcpConnState.192.168.10.101.3074.192.168.10.1.5431:-->closeWait(8)
tcpConnState.192.168.10.101.3078.0.0.0.0.2048:-->listen(2)
tcpConnState.192.168.10.101.3078.192.168.10.100.139:-->established(5)
```

Syntax: INTEGER { closed (1) , listen (2) , synSent (3) ,
Status: mandatory
Access: read-write
Reference:
Index:
Object ID: .1.3.6.1.2.1.6.13.1.1
Description:

```
 "The state of this TCP connection.  The only value which nay be set by a manag
ement station is deleteTCB(12). Accordingly, it is appropr:ate for an agent to
return a `badValue' response if a management station attempts to set this objec
t to any other value.  If a management station sets this object to the value de
```

Figure 14. SNMP Attribute TCP Sub-tree GET Operation Results

F. SUMMARY

This chapter describes the experimental design and testing result. There are only a handful of SNMP attributes we can change directly in order to control the network topology. This is the result of the SNMP design since it was supposed to be "Simple" from the beginning and do only "Monitoring." The function of control was left outside and therefore each hardware or software vendor could implement the control function separately and in a proprietary way. There are also some other weaknesses of SNMP protocol and these concern the information provided per connection. This way we cannot know through the SNMP monitoring what the bandwidth of a specific TCP connection is. We know, for example, the number of packets in and out per interface, but the data about specific connections are not provided.

The latter means that even if we had the capability to intervene and to change specific attributes in order to provide self-synchronization and self-configurability of the network, this cannot be done as we miss the data of specific connections.

In the following chapters we propose an architecture that will circumvent the inherent disadvantages of SNMP regarding the “Monitoring” and “Control” of a network topology.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ARTIFICIAL INTELLIGENCE AGENT TECHNOLOGY

A. INTRODUCTION

In order to establish a new architecture that is based on AI, we need to cite some basic concepts of the components that we use. Our new architecture will be based on several layers that are based on Artificial Intelligence technology. This is necessary as our network needs to be adaptive or, in other words, must be ready to "make the same decisions that a human being would make."

In the midst of an undefined problem and uncertainty, computers are not good at deciding what to do next. In traditional programs, every situation that a computer may encounter must be explicitly anticipated and coded by a programmer. In most cases, people happily accept computers as obedient, literal and unimaginative servants. However, for an increasingly number of applications, we need systems that can decide what to do on their own in order to satisfy their design objectives. Such systems can be built with the help of agents (Weiss, 1999). These agents are the core entities of agent-based models. They populate and interact with each other and the environment in these models (Axelrod, 1997). Agents are adaptive software devices. When combined in moderate to large numbers, these agents can produce decisions and behaviors that are rational, even in ill-defined or dynamically changing complex situations. These rational decisions and behaviors advance the agent toward achieving its goal or intentions.

The study of complex systems that have many actors and their interactions often becomes too complex for a mathematical model. Agent-based modeling is a tool to study this kind of system. The tricky part of this modeling tool is to specify the environment, agent-knowledge model and the interactions between the agents (Axelrod, 1997).

B. ARTIFICIAL INTELLIGENCE AGENTS

1. Definition

There is no universally accepted definition of the term "agent." The lack of such a definition is primarily because various attributes associated with agency are of differing importance for different domains. For some domains, learning is the most important aspect of an agent, yet it may be not only unimportant but also undesired for other domains. The only concept present in almost all definitions of agents is "autonomy" (Weiss, 1999). Agents have autonomy. This means that their actions are the result of commands obtained from a user, and the result of a set of goals and tendencies embedded in them (Farber, 1999). An agent can be any type of physical or software entity that fulfills the basic concepts of agency. Ferber defines the properties of an agent as follows:

- An agent is capable of acting and modifying its environment.
- An agent can communicate with other agents in the environment.
- An agent has intentions.
- An agent controls some local resources.
- An agent is capable of perceiving its environment (a reactive agent may not have any representation of its environment).
- An agent possesses skills and can offer services.
- An agent may be able to reproduce itself.

Examples of what an agent can represent include beings, organizations, vehicles, or nations. Agents have the ability to perceive the environment, which in turn affects the agent's decision process. These actions are embedded in the agent structure generally as weighted-rule sets. The weights of these rules are updated continuously according to their performances in the past. The rule with the highest weight determines the next movement of an agent. Some ineffective rules can even be replaced with new ones. This property allows agents to adapt to their environment more rapidly.

The ability to adapt to their environment is one of the most important properties of agents that distinguish agent-based modeling techniques from other conventional techniques. From the agent's point of view, adaptation means changing the rules of actions based on what the agent has learned from previous interactions. The adaptation capability of an agent allows a simulation to imitate the behaviors of increasingly complex systems.

An agent's internal mechanism for achieving intelligent behavior can range from quite simple (in the case of a reactive agent) to exceedingly complex (in the case of cognitive agent). Cognitive agents have some internal representation of the environment that they operate in. The sensory information from outside the agent is processed in the representation before taking a new action. Thus, cognitive agents can operate in a relatively independent way. By contrast, reactive agents do not have an internal representation of the environment. As a result, they take an action according to the information directly sensed from the environment or according to the internal motivations that prods them toward accomplishing the task. Since reactive agents are incapable of performing complex tasks individually, they are often deployed in large numbers to overcome this limitation. Figure 14 depicts the difference between cognitive and reactive agents.

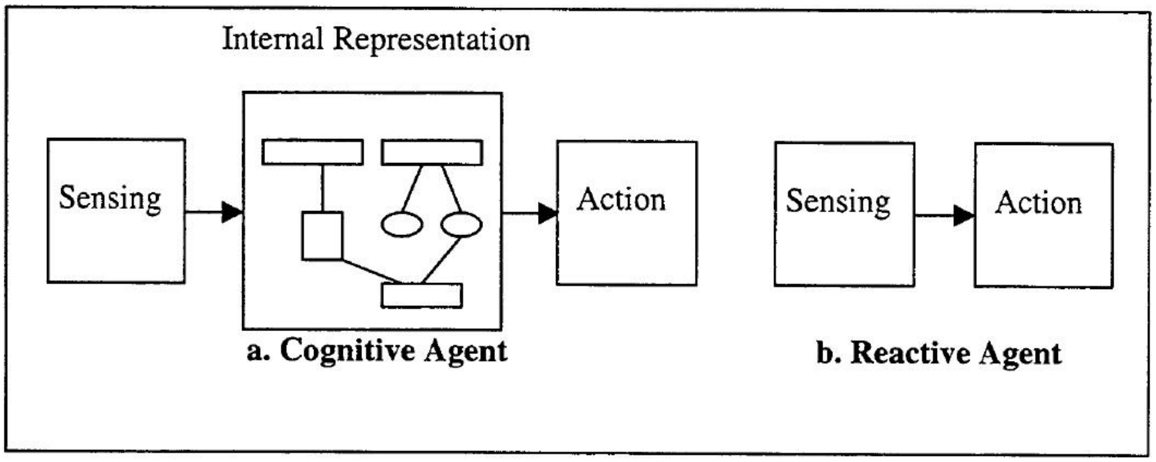


Figure 15. The Difference Between Cognitive and Reactive Agents

2. Multi-Agent Systems (MAS)

Just like the term agent, finding a widely accepted definition of MAS is difficult. Weiss (1999) gives the following characteristics of multi-agent environment: They provide a basis for specifying interaction and communication protocols; they are mostly open and have no centralized designer; and they contain autonomous and distributed agents that may be cooperative or self-interested. Instead of defining MAS characteristics, Ferber (1999) reports elements that comprise MAS. These elements are environment, objects, agents, relations, operations, and operations. Environment is a space in which every object of the MAS resides. Everything in the environment is an object. An agent is also an object in the environment that satisfies agency requirements. Relations link objects to each other in the environment. Operations are the actions that agents can perform in order to modify the environment and to achieve their goals. Operators can be described as the laws of the environment. Constructing MAS requires detailed models of these elements.

Moreover, Ferber defines four types of MAS according to the communication ability and physical existence of the agents in the environment. These are

- Communicating MAS: MAS in which agents are situated and have an ability to communicate with each other.
- Purely Communicating MAS: MAS in which agents are not situated but can communicate with each other.
- Situated MAS: MAS in which agents are situated and can communicate.
- Purely Situated MAS: MAS in which agents are situated but cannot communicate with each other.
- Ferber describes three levels of organizations studied in multi-agent systems:
- Micro-Social Level: Interactions between agents and the various forms of links are considered for this level.
- Group Level: Intermediary structures are considered for this level.
- Population Level: Dynamics of a large number of agents, together with the general structure of the system and its evolution are considered for this level.

3. MAS Simulations

Computer simulation imitates selected properties of reality, usually to predict the future or to practice and to rehearse problem-solving skills (Thinking Tools, 1999). The phrase “imitation of selected properties of reality” implies the model of the system under investigation. In most modern simulations, these models are based on either mathematical or rule-based relationships between system variables, which can be measured in reality. The most frequently used modeling methods are transition matrices, differential equations and rule-based “if-then” systems. These models are either deterministic or stochastic depending on the nature of the system under study.

Multi Agent Simulation is a new solution to the problem of imitating complex adaptive systems. Axelrod (1997) describes MAS simulation as “a way of doing thought experiments,” the goal of which is to enrich our understanding of fundamental systems. He contents that the goal of MAS simulation is not to find solutions to real world problems, but rather to provide insight into complex systems that conventional approaches cannot model. Therefore, modeling every aspect of the system is unnecessary. Axelrod (1997) proposes the famous army slogan “Keep it simple, stupid” (KISS) to the MAS simulation designers. Otherwise, the change in the outcome of the simulation cannot be linked to any particular variant in the simulation and hence makes simulation useless. However, one should also be very careful in deciding which aspect of the real world should not be included in the simulation. Omitting a key component of a system from the simulation may result in meaningless, undesired outcomes.

C. AGENT COMMUNICATION

The battlefield network is characterized by being highly distributed, heterogeneous, extremely dynamic, and comprising a large number of autonomous nodes. The basic requirements that have to be fulfilled are

- The emerging architecture of grid computing cannot be dealt with the traditional model of client-server architecture.
- Different platforms, services, operating systems, and standards have to be handled.

A community of intelligent agents can address these problems. Intelligent agents are agents that can communicate with each other, work together to accomplish complex goals, act on their own initiative, and use local information and knowledge to manage local resources and handle requests from peer agents.[11]

D. CASE BASE REASONING

1. Background and Motivation

Case-based reasoning is a problem solving paradigm that in many respects is fundamentally different from other major AI approaches. Instead of relying solely on general knowledge of a problem domain, or making associations along generalized relationships between problem descriptors and conclusions, CBR is able to use the specific knowledge of previously experienced, concrete problem situations (cases). A new problem is solved by finding a similar past case and reusing it in the new problem situation. A second important difference is that CBR is also an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems.

2. What is Case-based Reasoning?

CBR solves a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation. Let us illustrate this by looking at some typical problem solving situations:

- A physician, after having examined a particular patient in his office, recalls a patient whom he treated two weeks ago. Assuming that the recollection was caused by a similarity of important symptoms (and not the patient's hair color, say), the physician uses the diagnosis and treatment of the previous patient to determine the disease and treatment for the patient in front of him.

- A drilling engineer, who has experienced two dramatic explosions, is quickly reminded of one of these situations (or both) when the combination of critical measurements matches those of a previous explosion. In particular, he may recall a mistake he made during a previous explosion, and use this to avoid repeating the error once again.

[11] J. Mayfield, Y. Labrou, T. Finin, "Desiredata for Agent Communication Languages", Proceedings of the AAAI Symposium on Information Gathering from Heterogeneous, Distributed Environments, AAAI-95 Spring Symposium, Stanford University, Stanford, CA March 27-29

- A financial consultant working on a difficult credit-decision task recalls a previous case, which involved a company in similar trouble as the current one. He then recommends that the loan application be refused.

3. Case-based Problem Solving

As the above examples indicate, reasoning by re-using past cases is a powerful and frequently applied way to solve problems for humans. This claim is also supported by cognitive psychological research. Part of the foundation for the case-based approach, is its psychological plausibility. Several studies have given empirical evidence for the dominating role of specific, previously experienced situations (what we call cases) in human problem solving (e.g. [Ross-89]). Schank [Schank-82] developed a theory of learning and recall based on retaining experience in a dynamic, evolving memory2 structure. Anderson [Anderson-83] has shown that people use past cases as models when learning to solve problems, particularly in early learning. Other results (e.g. by W.B. Rouse [Kolodner-85]) indicate that the use of past cases is a predominant problem solving method among experts as well. Studies of problem solving by analogy (e.g. [Gentner-83, Carbonell-86]) also show the frequent use of past experience in solving new and different problems. Case-based reasoning and analogy are sometimes used as synonyms (e.g. by Carbonell). Case-based reasoning can be considered a form of intra-domain analogy. However, as discussed later, the main body of analogical research [Kedar-Cabelli-86, Hall-89, and Burstein-89] has a different focus, namely analogies across domains. In CBR terminology, a case usually denotes a problem situation. A previously experienced situation, which has been captured and learned in a way that it can be reused to solve future problems, is referred to as a past case, previous case, stored case, or retained case. Correspondingly, a new case or unsolved case is the description of a new problem to be solved. Case-based reasoning is – in effect – a cyclic and integrated process of solving a problem, learning from this experience, solving a new problem, etc. Note that the term “problem solving” is used here in a wide sense, coherent with common practice within the area of knowledge-based systems in general. This means that problem solving is not necessarily the finding of a concrete solution to an application problem. It may be any problem put forth by the user. For example, to justify or criticize a solution proposed by the user, to interpret a problem situation, to generate a set of possible solutions, or generate expectations in observable data are also problem-solving situations.

4. Learning in Case-based Reasoning

A very important feature of case-based reasoning is its coupling to learning. The driving force behind case-based methods has come from the machine learning community to a large extent, and case-based reasoning is also regarded a subfield of machine learning. Thus, the notion of case-based reasoning does not only denote a particular reasoning method, irrespective of how the cases are acquired, it also denotes a machine learning paradigm that enables sustained learning by updating the case base after a problem has been solved. Learning in CBR occurs as a natural byproduct of problem solving. When a problem is successfully solved, the experience is retained in order to solve similar problems in the future. When an attempt to solve a problem fails, the reason for the failure is identified and remembered in order to avoid the same mistake in the future. Case-based reasoning favors learning from experience, since it is usually easier to learn by retaining a concrete problem solving experience than to generalize from it. Still, effective learning in CBR requires a well worked out set of methods in order to extract relevant knowledge from the experience, integrate a case into an existing knowledge structure, and index the case for later matching with similar cases.

5. Combining Cases with Other Knowledge

By examining theoretical and experimental results from cognitive psychology, it seems clear that human problem solving and learning in general are processes that involve the representation and use of several types of knowledge, and the combination of several reasoning methods. If cognitive plausibility is a guiding principle, architecture for intelligence where the reuse of cases is at the center, should also incorporate other and more general types of knowledge in one form or another. This is an issue of current concern in CBR research [Strube-91].

6. The CBR Cycle

At the highest level of generality, a general CBR cycle may be described by the following four processes:

- RETRIEVE the most similar case or cases
- REUSE the information and knowledge in that case to solve the problem
- REVISE the proposed solution
- RETAIN the parts of this experience likely to be useful for future problem solving

A new problem is solved by retrieving one or more previously experienced cases, reusing the case in one way or another, revising the solution based on reusing a previous case, and retaining the new experience by incorporating it into the existing knowledge-base (case-base). The four processes each involve a number of more specific steps, which are described in the task model. Figure 16 illustrates this cycle.

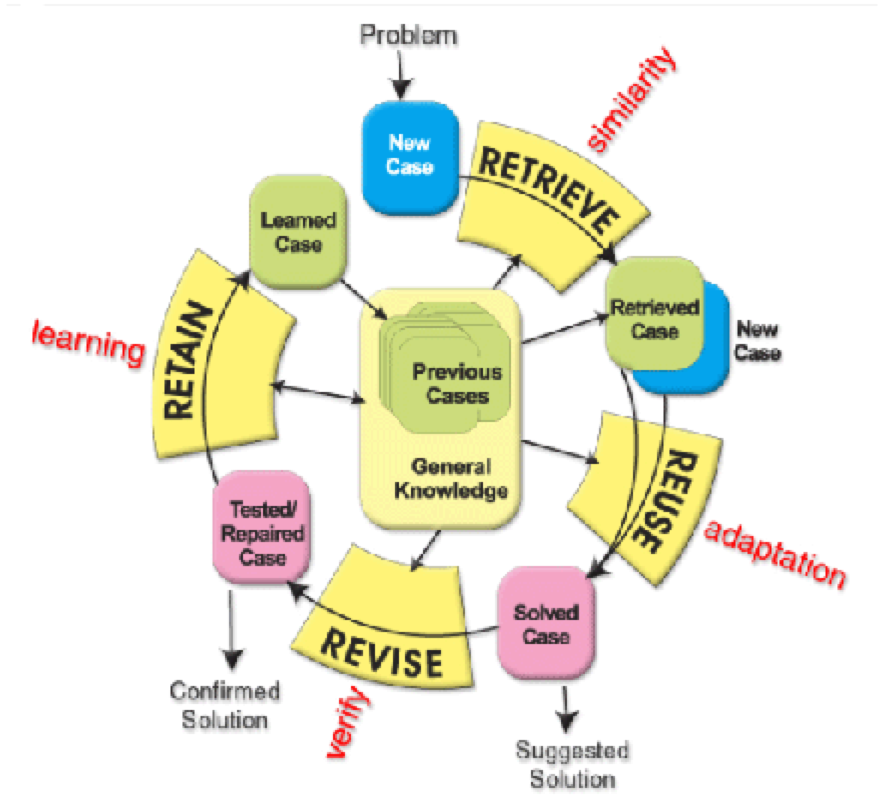


Figure 16. The CBR Cycle

An initial description of a problem (top of figure) defines a new case. This new case is used to RETRIEVE a case from the collection of previous cases. The retrieved case is combined with the new case – through REUSE – into a solved case, i.e. a proposed solution to the initial problem. Through the REVISE process, this solution is tested for success, perhaps by being applied to the real-world environment or evaluated by a teacher, and repaired if it failed. During RETAIN, useful experience is retained for future reuse, and the case base is updated by a new learned case, or by modifying some existing cases.

As indicated in Figure 16, general knowledge usually plays a part in this cycle, by supporting the CBR processes. This support may range from very weak (or none) to very strong, depending on the type of CBR method. By general knowledge, we mean general domain-dependent knowledge, as opposed to specific knowledge embodied by cases. For example, in diagnosing a patient by retrieving and reusing the case of a previous patient, a model of anatomy together with causal relationships between pathological states may constitute the general knowledge used by a CBR system. A set of rules may have the same role.

E. MOBILE AGENTS IN METWORK MANAGEMENT

1. Mobile Agents

Considering mobile agents in telecommunication networks entails several advantages. This includes reducing message processing between distributed entities and so limiting the bandwidth required for management purposes. Also mobile agents permit local processing that enables fast reaction to external changes. Lastly, the design of distributed applications in which management logic supposes the visit of several nodes that is a significant improvement in terms of scalability and robustness.

A very comprehensive study of mobile processing for network management was done within the Perpetuum Mobile Procura (PMP) project.[12] In the context of this project, a taxonomy of mobile codes lead to the definition of various kinds of mobile agents, which are supposed to evolve in the same infrastructure in order to fulfill different functional areas of management areas. These agents evolve in a particular mobile code environment (MCE), which is part of a network element and provides for example

[12] Andrzej Bieszczad, "Advanced Network Management in the Network Management Perpetuum Mobile Procura Project", SCE Technical Report SCE-97-07, March 1997.

migration, communication and security facilities, and an interface to managed resources within the network element.

2. Active Networks

Active Networks (AN) are based on a similar approach, namely on Active nodes containing execution environments for capsules transponding the Active Applications. The AN infrastructure therefore appears to be an ideal candidate for the implementation of mobile agent-based applications. It supports the transfer of code via capsules and also ensures reliable communication channels as well as a secure access to the local recourses. Furthermore, it supports several execution environments so that active applications can be divided into different categories according to their application domain. Still, the deployment of mobile agent systems into ANs is at an early stage and reveals several issues such as security, performance, safety, garbage collection, and platform independence.[13] All the above approaches are based on mobile agents that act more or less isolated – there is no notion of a society of mobile agents and little or no coordination or cooperation ability. A sample approach based on a population of cooperating mobile agents is the one proposed by the MIT.[14] The experiments were performed with a discrete simulation of a mobile ad-hoc network involving several types of mobile agents working cooperatively to build a network map that can be used for routing decisions. Here, coordination is based on direct communication when agents meet and exchange knowledge.

On the contrary, there are also approaches based on indirect communication. In systems endowed with the property Swarm Intelligence,[15] unintelligent agents with limited individual capabilities collectively exhibit intelligent behavior. This bio-inspired approach draws its model from evolutionary biology and from the study of ecosystems such as bacteria, or the societies of ants or bees: it turns out that these societies exhibit

[13] Stamatis Karnouskos, "Agent-populated Active Networks," Proceedings of the 2nd International Conference on Advanced Communication Technology (ICACT' 00), Summer 2000.

[14] Nelson Minar, Kwindla Hultman Kramer, Pattie Maes, "Cooperative Mobile Agents for Dynamic Network Routing." Chapter in book *Software Agents for future Communication Systems*, A.L.G. Hayzelden, J. Bigham (Editors), Springer, 1999.

[15] Eric Bonabeau, Marco Dorigo, Guy Theraulaz, *Swarm Intelligence – From Natural to Artificial Systems*, Oxford University Press, October 1999.

social coherence although the individual's behavior is mainly stochastic. This behavior is known as emergent behavior.

A very similar approach is AntNet,[16] a mobile multi-agent system aimed at the adaptive learning of routing tables in communications networks. The system refers to the Stigmergetic Concept, which may be defined as the indirect communication mediated by modifying environmental states that are locally accessible by the communication agents.

In the following chapter I propose a particular mobile multi-agent system architecture dealing with the issues of mobility, coordination, communication, and agent population in a transparent way.

F. SUMMARY

This chapter described the basic principles behind Artificial Intelligence Agents. The principle of agents is necessary for laying down our proposed architecture. This is because we need a layer of making the necessary decisions about the network configuration after it is fed with the proper information and has consided previous relative knowledge.

[16] Gianni Di Caro, Marco Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research 9* (1998) pp. 317-365.

V. ADAPTIVE NETWORK MANAGEMENT ARCHITECTURE

A. INTRODUCTION

Before delivering our proposed architecture, it is imperative to cite the course of action of our research up to this point.

First we introduced the idea of an adaptive network that can be configurable, adjustable, scalable, and well-balanced without human intervention. One seemingly easy way to implement this idea is to configure the values of the SNMP protocol that all the managed clients adopt in a network. At the test bench this idea proved to be not working, for the SNMP protocol has inherent simplicities and securities that prevent such direct intervention.

Now we will circumvent the difficulties of the SNMP protocol by trying to establish a two-way communication with the network layer. This can be done in one way by feeding the application layer with data from the SNMP protocol and by communicating and controlling the network layer from the application layer using TELNET with the routers. The latter communication can be wrapped into Java classes that need to be specific for each router model (Cisco, Linux, and Windows).

The MANagement, Testing and Reconfiguration of IP based networks using Mobile Software Agents (MANTRIP) project is a major contribution in establishing our architecture. Its outcomes have successfully implemented network control through an application layer. What this project misses is the ability to have a layer that will produce the necessary decision and transform the network from controllable to adaptive. In the next section, we briefly describe the MANTRIP project and provide an additional and necessary Artificial Intelligence layer in order to establish our proposal.

Dr Michalas, research Assistant at the Media Lab, National Technical University of Athens, Greece provided the code of MANTRIP project. Our experimentation with the code was very useful, but very restricted because the specific implementation for Cisco / Linux Routers were not available. Nevertheless, we did realize that the schema and the implementation of the MANTRIP project could be the base of our architectural proposal since it ideally circumvents ideally the SNMP communication problem we stated in Chapter III.

B. THE MANTRIP PROJECT

1. Introduction

The objective of the MANTRIP project is applying this newly distributed and autonomous software technology to manage IP networks. Experiences from the use of this emerging paradigm and technology in the context of MANTRIP are presented in this paper, which has the following structure. Section 2 presents a brief overview of MANTRIP system. Section 3, is the key section, presents in detail the agent-based MANTRIP QoS Management application. Finally, Section 4 presents our experience with agent technology in the context of QoS management in IP networks.

2. The MANTRIP Project

The MANTRIP project exploits the mobile agent features in order to provide novel management systems and applications. Each of the developed applications integrates a set of software modules in a common environment – a Mobile Agent Platform, either Voyager or Grasshopper. Using different applications, both the network operator/administrator and users can access management information on end-to-end performance, alarms, protocol conformance, QoS, etc. and can interact with the network in order to configure and reconfigure it when required.

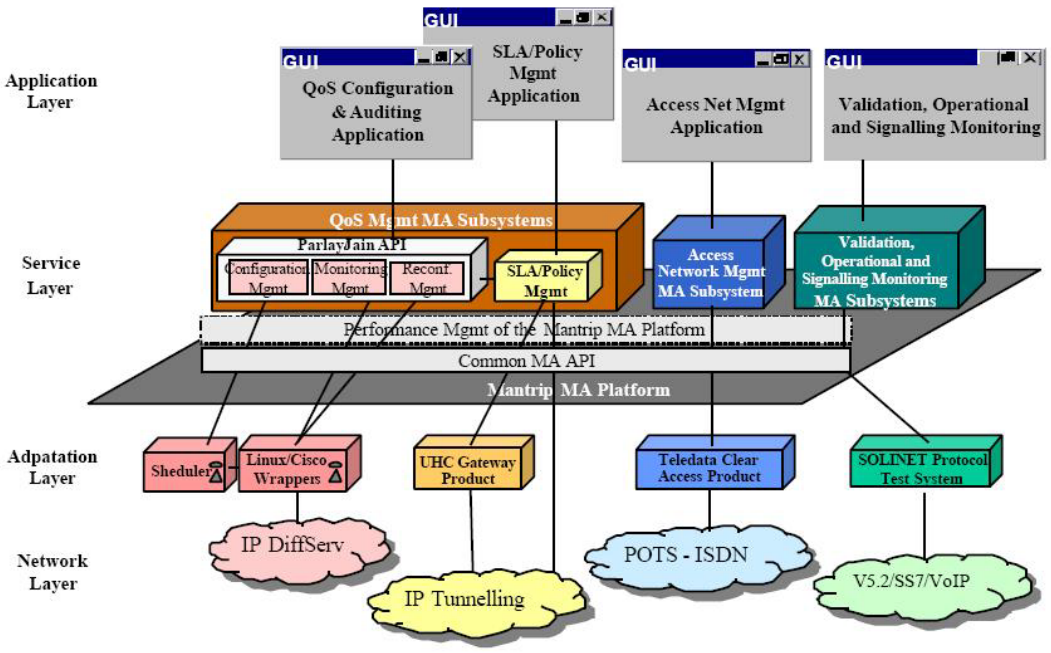

Application Layer
GUI
QoS Configuration & Auditing Application
SLA/Policy Mgmt Application
Access Net Mgmt Application
Validation, Operational and Signalling Monitoring
Service Layer
QoS Mgmt MA Subsystems
Parlay/Jain API
Configuration Mgmt
Monitoring Mgmt
Reconf. Mgmt
SLA/Policy Mgmt
Access Network Mgmt MA Subsystem
Validation, Operational and Signalling Monitoring MA Subsystems
Performance Mgmt of the Mantrip MA Platform
Common MA API
Mantrip MA Platform
Adpatation Layer
Sheduler
Linux/Cisco Wrappers
UHC Gateway Product
Teledata Clear Access Product
SOLINET Protocol Test System
Network Layer
IP DiffServ
IP Tunnelling
POTS - ISDN
V5.2/SS7/VoIP

Figure 17. MANTRIP Management System

The overall architecture is depicted in Figure 17. The different software modules that constitute the overall management system are organized in different layers of concern.

- The Application Layer includes both existing management applications and applications developed within the MANTRIP project. They are

1. Conformance Testing and Signaling Monitoring Application
2. Configuration and Alarm Management application for Access Networks Application
3. SLA Management Application and Policy Network Based Management
4. QoS Management Application, it allows dynamic configuration, monitoring and reconfiguration of QoS parameters in IP DiffServ Network resources.

- The Service Layer contains the software modules – services that support the execution of various applications. These are services like the QoS Configuration and QoS Monitoring as well as services enriching the MA platform's capabilities. As an example, the Common MA API abstracts away from the type of MA platform. Either Grasshopper or Voyager can be used without having any impact on the other services.

- The Adaptation Layer is responsible for hiding the protocol details from the service layer. It includes the network adapters and wrappers.

- The Network Layer includes the network resources and existing network management platforms to be managed by the mobile agent platform.

This layered architecture makes the overall MANTRIP management system more flexible.

On the other hand, Mobile Agents can be used to decentralize the processing of some of the most important management tasks. Other technologies may be used for this purpose, but the most common paradigms like client/server, may easily lead to poor performance when extensively used in management systems due to the amount of information that must be exchanged between the central and remote nodes. Decentralization improves the scalability of management systems. When networks of considerable size must be managed, this is definitely an important feature. Moreover, agents can easily be replaced in real time. Mobile agents in a remote fashion can perform heavy management tasks as network configuration, performance monitoring and alarm processing. When a new task needs to be accomplished, a new agent can be created and migrate to a remote place to perform its task without disrupting the normal functioning of the local system. Mobile Agents can drastically decrease the need for transferring big amounts of information to central stations.

In summary, agent technology, when properly used, can become a powerful tool to increase software flexibility, distribution and decentralization.

3. QoS Management Application

The QoS management architecture has been defined based on the following main requirements:

- Protocol and Vendor Independence: Introducing new types of routers, (e.g. Cisco, Linux, Redstone), eventually coming from different vendors, should not impact the applications.

- Network Technology Independence: The impact of introducing new IP-based technologies (DiffServ, MPLS) should be minimized.

- Openness and Flexibility: The interfaces between the different layers should be open, flexible, and standard whenever possible in order to promote quick development of management applications.

Based on these requirements, the MANTRIP project built an architecture that combines open and standard APIs with mobile agent technology. It tried to take advantage of both worlds. Figure 18 shows the QoS Management Architecture. In between the software modules represented below, standard and tailored APIs have been used.

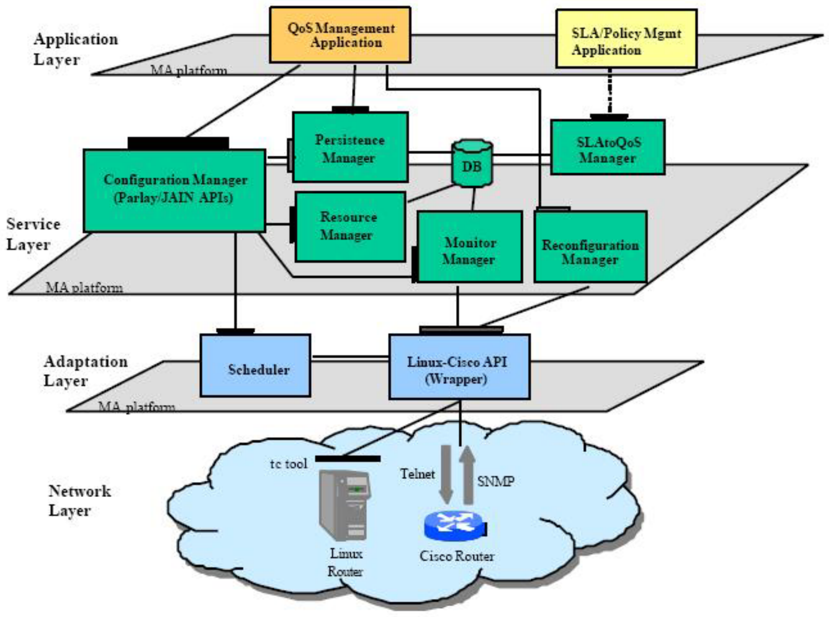


Figure 18. QoS Management Architecture

a. QoS Configuration and Monitoring

The QoS Management Architecture includes the following components and subsystems:

- The QoS Configuration and Monitoring Application offers a Graphical User Interface (GUI) to both network administrators and users. By using this GUI, administrators may profit from an extensive set of privileges: change dynamically the network configuration QoS parameters (e.g. max BW available per class of service, queue size, buffer size), monitor QoS of the traffic produced by users (delay, jitter, loss and bandwidth), change monitoring periods, create/delete users, access points, routers, create/delete and modify QoS templates, etc. The users can have access to all the information available in the GUI but they can not change it, they can only request uni or bi-directional resource reservation (i.e. set-up a scheduled VPrP – virtual provisioned pipe) by choosing a QoS template and certain value for the bandwidth, minor or equal to the maximum bandwidth available per class of service. The QoS templates made available to users may be constrained by the SLA defined externally by another application shown in the picture above, such as the SLA/Policy Management application.

- The Connectivity Management Subsystem implements in Java the Parlay Connectivity Manager API, an open and standard Application Programming Interface (API) specified by the Parlay Group. The implementation of this API includes a set of service sub-modules (Connectivity Manager, Persistence Manager and Resource Manager) that offer layer-generic IP connectivity management capabilities to the application. By using this subsystem, administrators can manage the network and populate the database with both application and network specific information while users can request or schedule resources and QoS. The Connectivity Management subsystem implements the mobile agents that are sent as closely as possible to the routers to perform the requested connectivity tasks.

- The Performance Monitor Management Subsystem offers users and administrator the capability of monitoring the QoS parameters of the configured connections. The Monitoring subsystem makes use of Active and Passive monitoring techniques for obtaining delay/jitter and bandwidth utilization/loss statistics respectively.

The user of this system also has the capability of modifying the granularity and report period of either monitoring service on-the-fly. This module implements the mobile agents that are sent as closely as possible to the routers to perform monitoring tasks.

- The Configuration and Reconfiguration Management Subsystem allows the administrator to (re-)configure certain QoS parameters in routers. On one hand, it caters to the initial configuration of the routers (i.e. define parameters for the DiffServ classes of service) and on the other hand, if a certain path flow is violating the thresholds defined by the QoS template, the administrator may trace the route of a certain VPrP (if static routing is being used), get the queue load of the routers involved and reconfigure them in order to improve the end-to-end QoS. Through this subsystem, it is possible to configure and reconfigure the Random Early Detection (RED) parameters, the queue size per class of service and the maximum bandwidth allocated to each class. Furthermore, the reconfiguration module estimates new bandwidth values according to both the queue load per class, and the required delay spacing among classes defined by the user. This module implements the mobile agents that are sent as closely as possible to the routers to perform (re-)configuration tasks.

- Scheduler and Wrappers include the static agents that are located as closely as possible to the routers in order to serve specific requests from visiting agents, which cannot access the network layer directly. A common tailored API (referred in Figure 17 as the Linux-Cisco API) has been defined to configure and to obtain QoS monitoring information (e.g. loss and bandwidth) from both Linux and Cisco routers. Due to the limited information that is possible to gather from the routers, some mobile agents are used to "emulate" traffic and measure some additional parameters (e.g. delay and jitter) in the context of Active Monitoring. Linux and Cisco wrappers implement this common API and hide from the upper layer the type of router being used in the network and the protocol details. The scheduler is located only in the edge routers and keeps track of the reservations and the resources available.

b. JAIN/Parlay API

A number of interfaces were considered for adoption and inclusion in the MANTRIP system, including the interfaces under development from: TSAS (OMG

Telecommunications Domain Task Force Activities), OSA API (part of the 3GPP standardization work), JAIN (Java API Initiative), Parlay API, Multi-Service Forum. Due to the functionality and status of the specification, the level of openness and the likely acceptance by the industry, the Parlay interfaces were adopted as the basis of inclusion in the MANTRIP system. A number of extensions, adaptations and clarification were made.

The Parlay Group is an open multi-vendor consortium formed to develop open technology-independent Application Programming Interfaces (APIs) enabling third parties to develop applications and technology solutions to operate across multiple networking platform environments. Most of the Parlay specifications have been submitted and accepted by standardization organizations, such as 3GPP and ETSI. Key operators and vendors such as BT, Telecom Italia, France Telecom, IBM, SUN, Nokia, Ericsson, Lucent, Alcatel, Siemens, Nortel and Cisco are members of the Parlay consortium.

The MANTRIP project adopted the Parlay Connectivity Manager API in order to configure QoS parameters in IP DiffServ Networks. This is an open API that may be used by any application developer (users, service providers, network operators, brokers, etc.).

Besides following the Parlay specification the project also decided to follow the approach suggested by the JAIN initiative. This group is supported by Sun Microsystems and has a similar objective with Parlay but more focused in scope, that is to define a set of open APIs in Java that abstract from specific network protocols and ease the fast development of next generation applications over a Java platform. As in JAIN, IP network management APIs have not yet been defined, so two levels of Java APIs were created. One at the network layer, the Linux-Cisco API, which abstracts from specific type of routers and protocols, and another one at the service layer, a Java implementation of the Parlay API, which offers to applications common access to network capabilities. A Java Parlay API may be used for local access while a Parlay CORBA API is highly recommended for application remote access.

c. Agent Interaction – Network Resources

In the MANTRIP prototype the agents have been placed as shown in Figure 19. In this network there are both Linux and Cisco routers that constitute the IP DiffServ environment.

The static agents are responsible for directly controlling the routers. They are called static because they migrate once to the routers, or as closely as possible to them, and they stay there until new functionalities need to be provided and they need to be replaced. These agents interact with the network elements in order to perform the tasks that the mobile agents tell them to do. They offer the Linux-Cisco API toward the mobile agents that will be using their services to configure, monitor and reconfigure QoS parameters in the DiffServ routers.

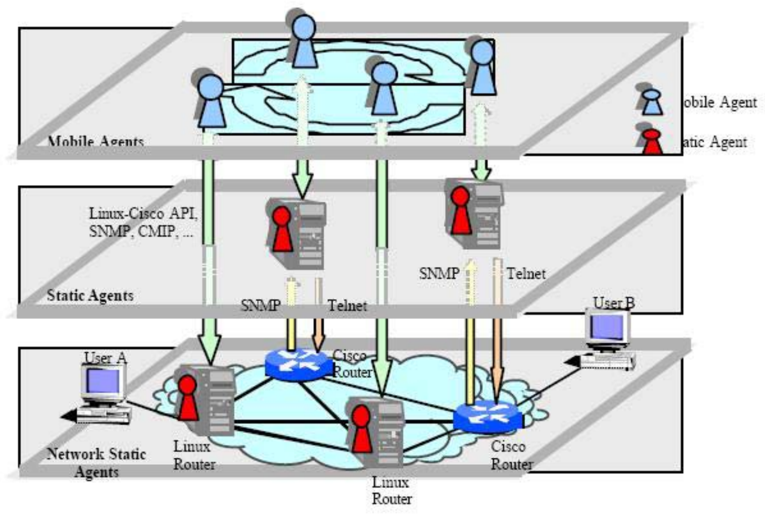


Figure 19. Agent Interaction

The wrappers are static agents, which offer the following set of interfaces:

- Edge_ConfiguringSLS: This interface allows for edge routers configuration. In other words it allows for configuring the classifiers and meters according to a SLS. All entries in the classification table can be retrieved. Overlapping filters are allowed, provided that they have different priorities. However, it is not the rule of the API implementation module to verify overlapping of filters.

- Monitoring: This interface allows one to monitor the amount of traffic forwarded and dropped by each class. It should be used to get the so-called Passive Monitoring Measurements, i.e. Packet Loss and Throughput.

- Core_Configuring: This interface is invoked by the administrator in order to provide a basic configuration to all routers, Core and Edge.

The Scheduler is also a static agent used to manage the time that the connectivity should be active according to the user requests (or SLA). It contains one interface:

- IScheduler

The mobile agents are created by the service subsystems and migrate to places where the static agents reside.

d. IP DiffServ Network Configuration

As mentioned before, the QoS management prototype operates on an IP DiffServ network. By using the Linux-Cisco API, it is possible to configure the internal modules of each DiffServ node, as represented in Figure 20. The filters of a router are composed by two modules: a classifier that identifies the traffic flow and a meter that characterizes the flow in terms of bandwidth. The meter supports TCM marking (Three Color Marker – Green, Yellow, and Red). The marker is responsible for effectively marking the IP packets by assigning a certain DiffServ Code Point (DSCP) to each packet according to its QoS characteristics. The DSCP is just a binary code, which is placed in the Type of Service (ToS) field of the IP header. The Per Hop Behaviors (PHBs) are defined by both the shaper/dropper and the scheduler. By configuring these modules, we define how each DSCP is used by the routers to select which packets should be forwarded first to the next router. Besides the traditional Best-Effort, there are two other PHBs: Expedited Forwarding (EF) and Assured Forwarding (AF). These behaviors differ from each other in terms of QoS guarantees. Distinct configuration of the waiting queues must reflect this difference.

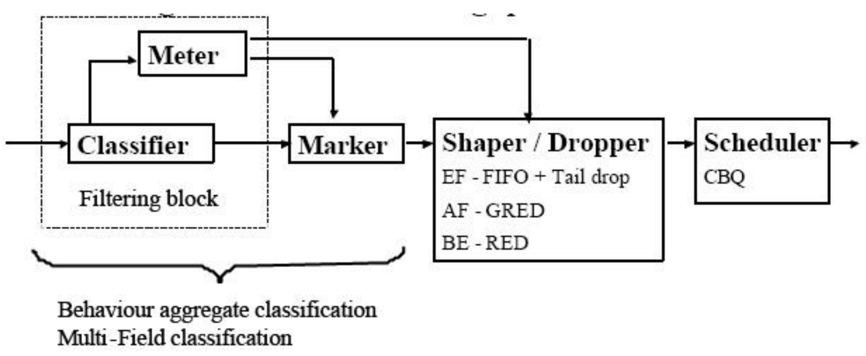


Figure 20. IP DiffServ Node

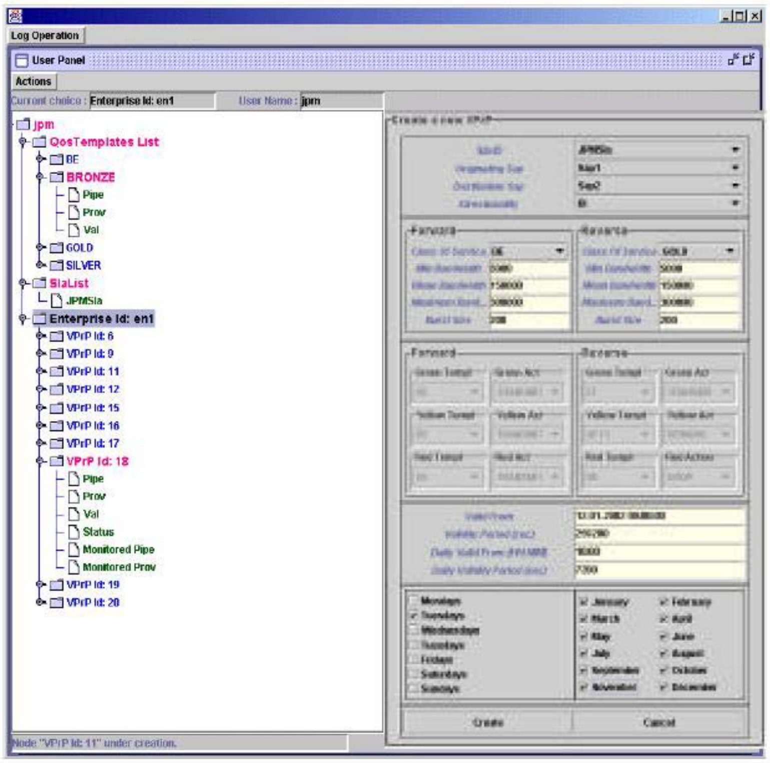
In order to implement the EF PHB, where the objective is to ensure a class of service with a minimum guaranteed bandwidth (configurable) and low losses, jitter and delay, the FIFO (First In, First Out) queuing mechanism has been used; packets are delivered in the same order in which they arrive to the queue.

The PHB AF defines four classes of service, each of which contains three subclasses (AF11, AF12, AF13, AF21, AF22, AF23, AF31, AF32, AF33, AF41, AF42, and AF43). The difference between them consists on the priority given to the packets that must be dropped. The AF DiffServ PHBs have been implemented using the Random Early Detection (RED) queuing mechanisms, where packets are dropped according to both queue occupancy and congestion probability.

Traffic from all the queues flows into the module that assigns the packets to be sent (router scheduler). This module was implemented by using Class Based Queuing (CBQ).

e. The QoS Management Application – Use Cases

The Graphical User Interface of the QoS management application is depicted in Figure 21.



Current choice : Enterprise Id: en1   User Name : jpm

```
jpm
  QosTemplates List
    BE
    BRONZE
      Pipe
      Prov
      Val
    GOLD
    SILVER
  SlaList
    JPMSla
  Enterprise Id: en1
    VPrP Id: 6
    VPrP Id: 9
    VPrP Id: 11
    VPrP Id: 12
    VPrP Id: 15
    VPrP Id: 16
    VPrP Id: 17
    VPrP Id: 18
      Pipe
      Prov
      Val
      Status
      Monitored Pipe
      Monitored Prov
    VPrP Id: 19
    VPrP Id: 20
```

SlaID	JPMSla
Originating Sap	Sap1
Destination Sap	Sap2
Directionality	Bi

Forward		Reverse	
Class of Service	BE	Class Of Service	GOLD
Min Bandwidth	5000	Min Bandwidth	5000
Mean Bandwidth	150000	Mean Bandwidth	150000
Maximum Band...	300000	Maximum Band...	300000
Burst Size	200	Burst Size	200

Valid From	12.01.2003 00:00:00
Validity Period (sec)	[illegible]
Daily Valid From (HHMM)	1000
Daily Validity Period (sec)	7200

Mondays
[✓ Tuesdays]
Wednesdays
Thursdays
Fridays
Saturdays
Sundays

[✓ January] [✓ February] [✓ March] [✓ April] [✓ May] [✓ June] [✓ July] [✓ August] [✓ September] [✓ October] [✓ November] [✓ December]

Create   Cancel

Node "VPrP Id: 11" under creation.

Figure 21. QoS Management Application: Configuration GUI

- Configuration Scenario: By using this GUI, the user can make connectivity reservations by choosing values for the following set of parameters:

  1. Class of Service (Bronze, Gold, Silver, Best-Effort, or others – Depends on QoS templates available).
  2. Service Access Points (SAPs) origin and destination.
  3. Directionality.
  4. Minimum guaranteed bandwidth.
  5. Time scheduling.

Taking the example presented in Figure 21, the user "jpm" is interested in using the network resources, every Tuesday between 10 pm and 12 pm, starting from 2003 December 12. Since they would like for example to watch a movie, they are interested in a good quality of service in one direction but not as good in the opposite direction. Therefore they request a bi-directional VPrP between SAP1 and SAP2 and choose BE in the forward direction and Gold in the opposite direction. Furthermore they know that the movie will not need much more than 2Mb/s. So, they request to use this bandwidth that is a percentage of the total one available for the Gold class of service. This means that the traffic below 2Mb/s, i.e. the well-behaved traffic, will be marked as Green and will have the highest priority (mapped into the EF DiffServ class of service). The traffic, which exceeds this value by 10%, will be marked as Yellow (mapped into one of the AF DiffServ classes of service). If it exceeds more than 10% it will be marked as Red (BE) and will have the lowest priority. When the day comes, the routers will be configured according to the user requirements. During the activation time the users will be able to monitor QoS of the traffic being sent. They will also be able to see if the contract is being violated with respect to the QoS requirements.

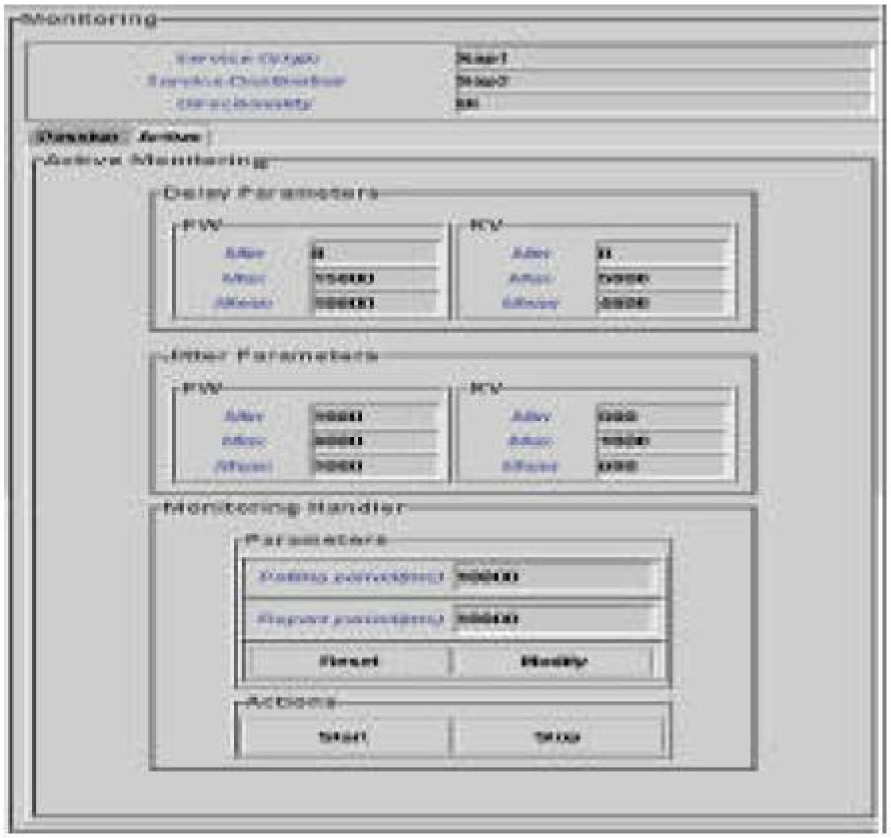
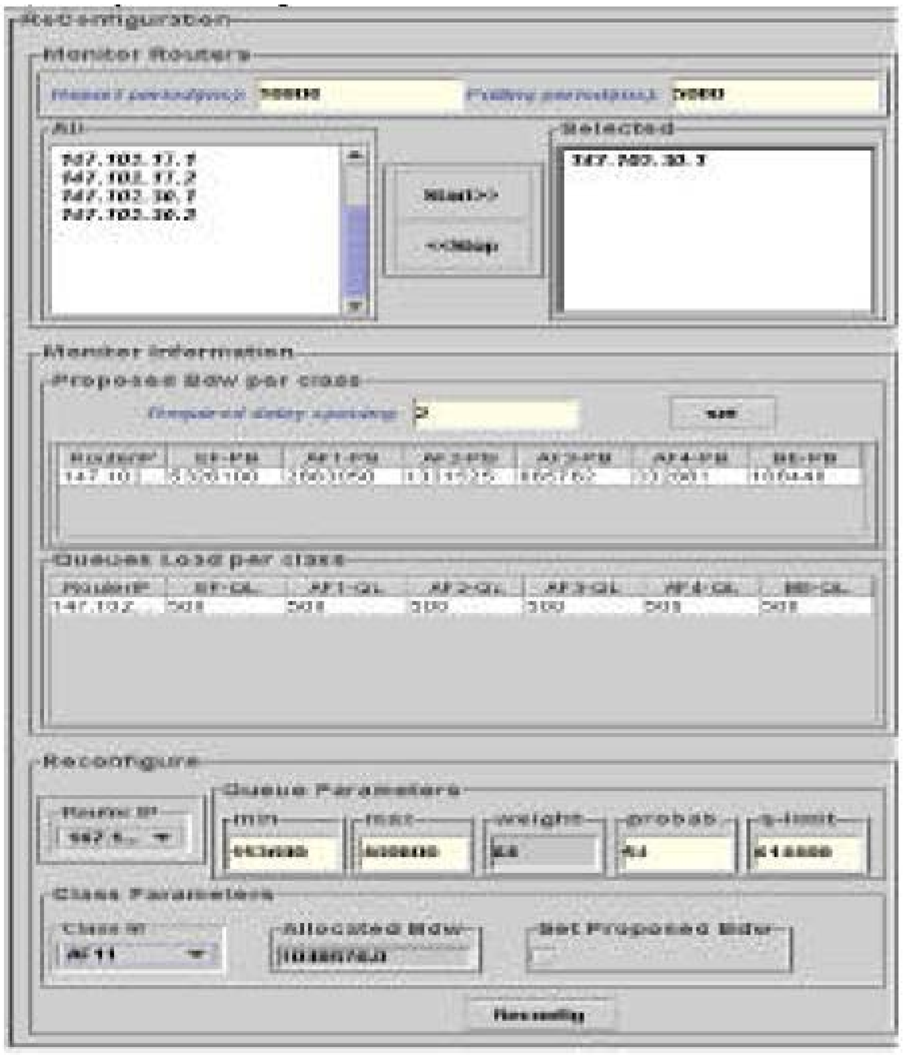


Figure 22. QoS Management Application: Managing GUI

- Monitoring Scenario: Through the monitoring interface (Figure 22) the user may ask to monitor the VPrP and observe how the QoS parameter values change. Performance monitoring tasks involve both active and passive measurements. Active measurements are taken by inserting (low-impact) test streams (delay, jitter, etc.), while passive measurements are taken by analyzing the traffic passing from a network element (used bandwidth, loss, etc.). The system provides performance reports and notifications in a scheduled and "on-the-fly" manner, respectively.



Monitor periodicity: 50000 Polling periodicity: 5000

All:
147.102.17.1
147.102.17.2
147.102.30.1
147.102.30.2

Selected:
147.102.30.1

Requested delay spacing: 2

RouterIP	EF-PB	AF1-PB	AF2-PB	AF3-PB	AF4-PB	BE-PB
147.102	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

RouterIP	EF-QL	AF1-QL	AF2-QL	AF3-QL	AF4-QL	BE-QL
147.102	508	508	500	500	508	508

Router IP: 147.1...
min: [illegible] max: [illegible] weight: [illegible] probab: [illegible] q-limit: [illegible]
Class Id: AF11 Allocated Bdw: [illegible]

Figure 23. QoS Management Application: Reconfiguration GUI

- Reconfiguration Scenario: Administrators can log on the application and get all the router interfaces involved in a VPrP by using the trace_route functionality. They can enter the reconfiguration screen, as depicted in Figure 23, select one or some of the traced IP addresses and get the RED parameters, the queue size and the actual queue load per class of service. The users can then reconfigure the router RED parameters and the queue size per class of service. Optionally, they can select the proposed bandwidth value per class of service given by the management system. After reconfiguring the administrators can go back to the monitoring GUI to check if the QoS parameters improved and, specifically, if the required delay spacing among the reconfigured classes is achieved.

4. Summary

The MANTRIP project is a management system based on mobile agent technology that supports quality of service management in IP networks. Using mobile agents as the basic design and implementation technology helped to decentralize configuration and monitoring tasks. In addition, it promoted good software design and made the implementation of such a complex system relatively easy. The savings for complex configuration and monitoring tasks are significant in comparison to client/server protocol-based or distributed object technologies. But most important, mobile agent technology supported the "programmability" of network nodes for configuration and monitoring purposes in a rapid manner, assuming only raw management capabilities available.

B. DESCRIPTION OF THE NEW ADAPTIVE MANEGEMENT PROTOCOL

1. Introduction

The previous section referred to the MANTRIP project. The main disadvantage of the MANTRIP project is that it cannot stand as an autonomous application. Although it serves the vertical distribution of a possible adaptive network, it does so up to the application level. From that layer, a physical administrator is needed to configure the network in order to get the maximum out of it. This is the reason that this project has so many interfaces (GUIs), so as the network administrator can monitor at real time the network characteristics.

This is not a viable solution for a very unstable, by nature, network such as the battlefield network. Even the establishment and continuous manning of a Network Operation Center is not enough for the continuously changing variables of the network. Wireless PDA's linked with satellite communications with servers and remote sensors require some hierarchy of their bandwidth needs. And this is only one small example of how volatile the network environment can be.

2. The Objective

Our objective is to propose a software architecture that will be sufficient to handle the management requirements of the next generation adaptive network. This low level software architecture will constitute a low-level layer of abstraction between the network layer and the network administrator application/person. This will serve as the substitute of the current SNMP protocol, and it will be an interoperable protocol across many platforms.

3. The Requirements

The first requirement that emerges from the objective is for the new protocol to be interoperable. We next cite all the possible requirements that are needed for a first implementation of the protocol:

- **Interoperable**. This protocol need to be the basic management tool for the many network schemas and operating platforms that currently exist. It has to cooperate fully with the layer of TCP/IP that stand underneath it and provide the necessary abstraction with any proprietary layer of control above it.

- **Controllable**. There is a great need for the new architecture to permit the intervention either of a user or through a software agent to change the basic parameters of the operating status of the network either at the link or at the router level.

- **Provide detailed data on network operation**. Current network management protocol SNMP provides a very generic picture of the network. Among its "deficiencies" is that it does not provide QoS (bandwidth, rate, loss rate) data for each connection (IP address, port) but only per interface (IP address).

- **Easy to implement**. The obstacle is that the implementation as the main infrastructure of a network is not ready to implement such a revolutionary protocol. We need to make it seem as an extension and/or advancement of the current SNMP protocol that could stand also backward compatible with the previous implementation of management applications.

C. STRATEGIC CONCEPT OF THE ADAPTIVE MANAGEMENT PROTOCOL

1. Generic Overview

We will implement the new architecture in the lower level of the network stack. This position entails many interfaces with the upper and lower levels of network abstraction. In Figure 24 there is the general concept upon which we base our implementation. As we can see, there are direct connectors with protocols like TCP, IP, and hardware like routers, servers, and operating systems. There are also significant dependencies on proprietary operating systems, such as Cisco IOS that presently rule the field of network middleware.

There is always the need for the correct interface of our new architect with the legacy systems. As the legacy systems present a great deal of simplicity that makes them more attractive than the complex and seemingly difficult and dangerous to implement new protocol, we need dependable interfaces that will ease the communication and the correct and fast data exchange of legacy with the new.

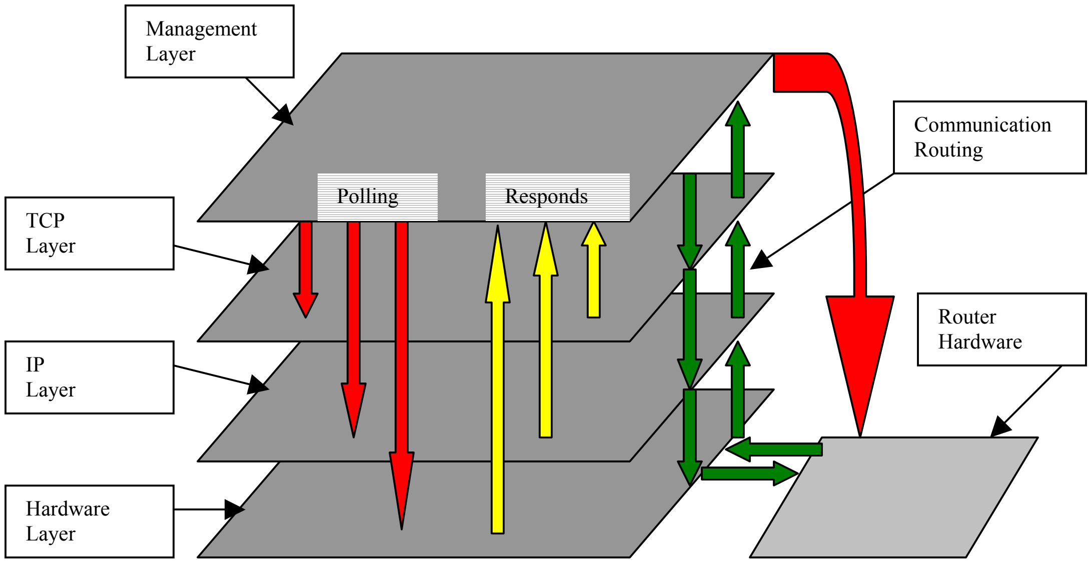


Figure 24. The New Adaptive Management Protocol.

2. Detailed Interconnections

The new Software Architecture Layer is required to establish all the connections the legacy SNMP had provided. This means that in addition to being enhanced, it must also be backwards compatible. As a result, we have to provide the following in order to achieve the best cooperation of the new with the legacy:

- **Hardware Compatibility**. The hardware that constitutes a network should have the appropriate software interfaces to provide the necessary functionality to the manager layer. What we need is increased access to routing information with a capability to create, delete, modify, and read the status of a connection that starts or ends within this middleware. For example, in the case of a router, we need to know, exactly, what are the QoS attributes per connection and not per interface (as we did in the legacy SNMP). We also must be able to modify the status of the connections. At this moment, no hardware manufacturer supports these operations. But they have to, at least in military systems. At this level of applications, the Commercial-Off-The-Shelf products cannot be so commercial.

But implementing this quality attribute to the new schema is most difficult. Manufacturers of network middleware control their products as highly valuable secrets. As the previous SNMP protocol was not so specific on how compatible the hardware middleware should be, the manufacturers established their own way of controlling their devices. Generally, these proprietary operating systems provide sufficient control such as the case of Cisco routers. But how can we incorporate such routers in our new schema? One way is to create wrapper classes from Java or C++ that can communicate with the router IOS through Telnet Protocol. This is an extremely difficult solution to implement as each router can listen to thousands of instructions and possibly millions of combinations of instructions. In this case, the wrapper class would be in the magnitude of 4 KLOC. This enormous software artifact is sufficient not for a family of routers but for only a specific router.

One possible solution is implementing hardware interfaces that do the job of software, faster and easier. Naturally, when we incorporate a new block in our diagram, then we expect delays. Nevertheless, these delays are insignificant when compared with the achievement.

- **Smooth Interaction with the TCP Protocol**. The management protocol stands virtually on top of the TCP layer. Although it does not communicate using TCP sockets, it uses the same libraries API that the operating system uses to interact with the network layer.

These legacy libraries are very restrictive in nature. Generally, the systems come preconfigured to support specific instructions and the core of this configuration is exactly this API. For example, in the case of Microsoft Windows 2000/XP family of operating systems, this API is the W32_sock.dll. There is also a specific SNMP agent that can support a handful of MIBs. But this is not enough. For our architecture to flourish, we need a separate API that will handle the UDP datagrams that our new architecture will use. This is because the needs are so huge and we have to break the libraries in more adaptable small parts, and because the platforms that we will implement upon our architecture are so specific in nature (military) that we can implement the new API as a supplement to the existing model.

- **Data Extraction from IP Layer**. We need detailed information from this specific layer. This information cannot be provided by the current IP technology, not even by the hardware interface technology. The new IPv6 protocol comes with amplified features that could support our new architecture. It supports QoS attributes and variables. Also it resolves the network addressing problem that became so apparent with the explosion of the World Wide Web. There are also some new characteristics such as detailed information per connection and statistics that would resolve our problem. But this new IPv6 has a long way to go before becoming the prevailing standard on the IP protocol. Until then and of course in the case the IPv6 fails due to national, commercial or individual interests, we have to explicitly declare a new standard that conforms to the requirements of our management architecture.

- **Secure Interactions through Operating Systems**. The heart of our solution is the most insecure computer creation ever made. The fact that an application has the ability to wander freely around the network nodes and make whatever changes it wants, is frightening at least. Maybe the network will be a dedicated for military interconnection, but what is more attractive to a hacker than an environment that when you succeed in breaking in, you have total control of the network. For this reason, security is another requirement that has to be fulfilled for the environment in order for our architecture to work without problems.

This requirement has to be implemented at the Operating System Level. There are programming languages, like Java and protocols like RMI, that ensure the spread of a mobile software agent without the worry of a possible hack. The security issue is resolved with special servers (mostly software servers) that act as an active registry for all the possible services and agents that are active at the time in our network. Also, there are special "signatures" that each agent has to bear in order to be eligible to act freely within a node.

D. COMPONENTS OF THE ADAPTIVE MANAGEMENT PROTOCOL

1. Generic Overview

At this stage, we examine in detail the basic components and their interactions of our new protocol. It is definitely these interactions that will define the final capabilities and the deficiencies.

Generally speaking, the new protocol will have the same characteristics as the legacy. Only it will be more detailed, complex, and more interconnected. The new protocol will be connectionless, easy to implement, and without special memory requirements.

2. The components of the new architecture

The following characteristics will give the new protocol the necessary functionality, as established in the requirements:

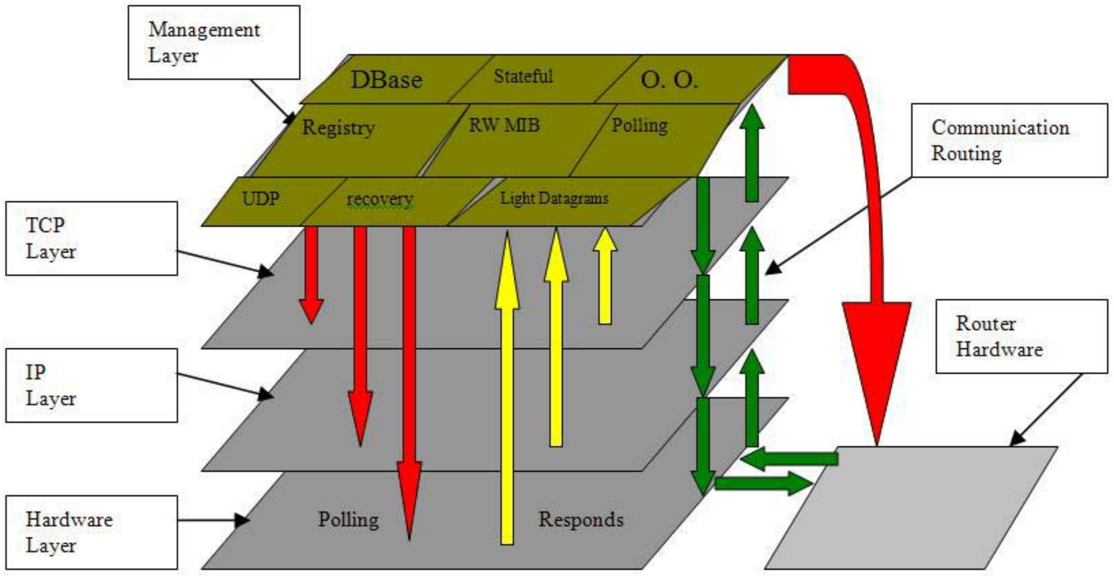


Figure 25. Components of the New Adaptive Management Protocol

- **UDP Connectionless Oriented**. Although UDP datagrams do not provide reliability, they have one great advantage over the TCP connections: They do not produce so much overhead. Neither do they reserve any bandwidth for their circulation. In this way we can manage a network with the least possible resources. When we deal with a noisy environment, we do not have the availability of resources even for UDP where this protocol might overload our network, as the volume of information is larger than the legacy SNMP. Nevertheless, the advantages of implementing this architecture may oversee a possible network overload.

- **Polling Based Data Acquiring**. This protocol will acquire the data needed by polling the managed stations. The polling can be done by the manager station or a dedicated station that will send the information to the manager. This polling might create an overhead depending on how large the network is. Also the legacy TRAP instruction will be available. This way a managed station will still be able to notify the manager of important issues. Only the new TRAP instruction will have to be enriched and will include i.e. possible happenings on a connection, connection instability, and possible need to allocate bandwidth to another connection.

- **Light Datagrams**. Datagrams not very long. Since the data that will flow using PDU are going to be larger in size, there is a need for a mechanism that will keep the length of the PDU small and prevent the overflow and still keep track of a possible PDU loss that will destroy the information. There is also the case that the new PDU might have the capability to carry instructions that can act directly on the managed station without any need for compilation or class downloading. Any possible PDU enrichment amplifies the network congestion and any attempt to lessen it in magnitude lessens its capability to carry more information.

- **Object Oriented**. This will enable the extraction of a whole object instead of constantly polling the tree as in the SNMP. In the context of the legacy SNMP, when we needed a table of data, we had to poll all the leaves of the specific branch sequentially. This would create enormous network overload, especially when the information we need is quite large. The extraction of an object, even as a serializable object will resolve this legacy problem and permit the efficient operation of the network.

- **Database Availability**. Instead of a MIB, it would be better to build a small Database with critical information that would help the manager application or the user. This can be critical attributes that by experience lead to the notion of an unstable network. Of course the notion of the database incorporates the need of a space in some disks and the danger that the next station that will resume management in case of a failure might not have the availability of this database. A possible solution might be the database to be very small in time and specimen depth and vary specific and adjustable to the manager's needs.

- **Fast Service Registration**. Enable fast and easy registration of services within the network and the easy de-registration. This will ensure that the users will have the ability to access the desired service quickly but also will contribute to the stability of the network as the services will facilitate the correct and fast implementation of software agents across the platforms and nodes.

- **Fast Users Registration**. This will make the transition to another managing node or another trapping report node in case of a failure easier. Also the nodes registration will facilitate the knowledge of the manager about the total network "population." Some special protocols, like Wireless IP require fast logging on/off services as their inherent instability tends to make a network one of the most difficult to manage. Another feature is the fact that the exact knowledge of the network topology will permit the design and population of routing tables. In this way, the network will have the sufficient resources for a faster adaptation.

- **Stateful**. Keep alternate routes in memory, so a possible re-routing would be fast. This requires the new protocol to be stateful and not stateless. The current network management protocol does not have the ability to "remember" successful routes; especially in the case one specific node "leaves" the network. These tables can be useful for a specific MAC address in a specific geographic location. This can be the product of a thinking procedure within the managing station (AI Agent).

- **Read/Write MIBs**. Creation and implementation of new MIBs that enable the READ and WRITE of an attribute. Also the agent should be able to change the new attribute down to the specific layer of abstraction (either it is TCP, IP, or hardware). This way, the MIB would act less like a pseudo database but more like a "registry." There, the new management protocol can refer often to see whether its operation status is conforming to the values of this "registry." In the case of a mismatch, the management protocol would be able to generate new SET datagrams that would bring the network to the desired condition.

E. POSSIBLE EVALUATION OF ADAPTIVE MANAGEMENT PROTOCOL

1. Introduction

To evaluate the proposed architecture, we use the Software Architecture Analysis Method (SAAM) as described by Clements et al. We choose this method because we are concentrating on the qualities and proposed capabilities of our architecture. The main steps include developing specific scenarios, a small description of the architecture, and a classification and prioritization of the scenarios.

2. Develop Scenarios

- **Scenario A.** Control the bandwidth of multiple simultaneous connections among multiple clients with one or more managing stations.

- **Scenario B.** Evaluate the percentage of network overload due to the circulation of managing datagrams. How does this affect the management of specific QoS attributes?

- **Scenario C.** Evaluate the security of the managing protocol by trying to impose malicious executable code inside the agent applications. As the new protocol will permit unrestricted access to the administrator by design, what would happen if the administrator gets hacked?

- **Scenario D.** Evaluate the scalability of the protocol by randomly changing the network topology or introducing new protocols like wireless IP where the rate of UDP losses are greater. What would happen in an environment full of all possible interference such as a battlefield, where we would like to deploy and manage a couple of hundreds PDAs sufficiently?

- **Scenario E.** Evaluate the possible evolution of the protocol to support a newer version of TCP, IP, or network topology such as for example IPv6 or even the OSI layer for networks.

- **Scenario F.** Explore any possible extension the protocol could accept in order to manage applications and not/in parallel with hardware connection.

- **Scenario G.** Develop an application that could possibly use past knowledge gained from experienced administrators and/or from evolvement of an Artificial Neural Network that could be used sufficiently and efficiently for fast and correct stabilization of an unstable by nature complex organization[17] such as a battlefield network.

3. Classify and Prioritize the Scenarios

The scenarios A, B, C, and D are classified as direct because they are satisfied by the architecture through the execution of the system. They will correspond directly to the requirements that were set at the design process and will increase the understanding of

[17] The term organization is used in order to show the magnitude of the enormous complexity that reigns in the boundaries of perfect operation and awful disaster, of a system such as internet and/or internetworking.

the system at the time of the testing. Also the main testing objects are the quality and performance attributes of the architecture.

These tests may lead to some additional iteration that would change, add, or even delete some system requirement that will be proved as impossible or even difficult to implement.

The last three scenarios E, F, and G are considered as indirect because they refer to possible evolutions of the architecture in order to be satisfied. They also address the scalability of the architecture, and how it can be combined with other technologies in order to provide advanced products. These change-cases will be posted as directives for a possible evolution or even integration of the architecture with other technologies.

In the limited available time for testing, we propose the following scenarios as the most important and with the highest priority in order to have an adequate view of the proposed architecture:

- **Scenario A.** Control the bandwidth of multiple simultaneous connections among multiple clients with one or more managing stations.

- **Scenario B.** Evaluate the percentage of network overload due to the circulation of managing datagrams. How does this affect the management of specific QoS attributes?

- **Scenario C.** Evaluate the security of the managing protocol by trying to impose malicious executable code inside the agent applications. As the new protocol will permit unrestricted access to the administrator by design, what would happen if the administrator gets hacked?

- **Scenario G.** Develop an application that could possibly use past knowledge gained from experienced administrators and/or from evolvement of an Artificial Neural Network that could be used sufficiently and efficiently for fast and correct stabilization of an unstable by nature complex organization such as a battlefield network.

- **Scenario E.** Evaluate the possible evolution of the protocol to support a newer version of TCP, IP, or network topology such as the IPv6 or even the OSI layer for networks.

F. ARTIFICIAL INTELLIGENCE LAYER INCORPORATION

One possible solution to the problem of the adaptive network, after having faced the two-way communication problem with the network layer through the new Adaptive Management Protocol, is adding one more layer on top of the architecture Project MANTRIP proposed.

This layer is the Artificial Intelligence Layer. It contains the necessary number of Artificial Intelligence Agents that will be fed with the data provide by the Mobile Agents Layer and will substitute the Application Layer. A more detailed view can be seen in Figure 26.

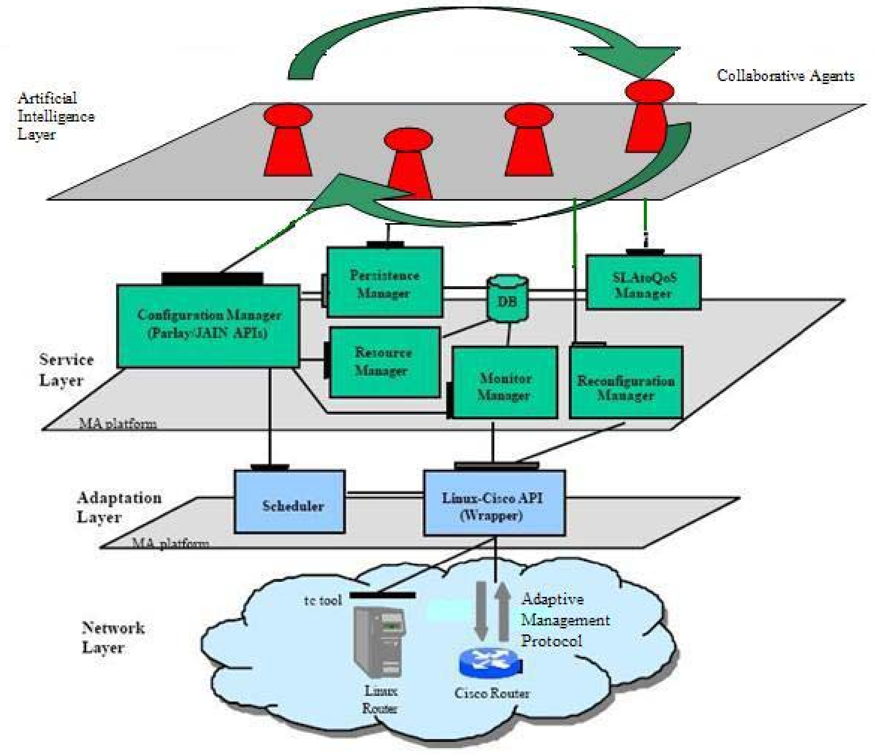


Figure 26. Proposed Architecture for an Adaptive Network

We can describe the Artificial Intelligence layer as comprising a set of Agent-facilitators[18]. These agents are used to extract a decision individually and act as a set in the context of a Multi-Agent Simulation in order to make the desired decision to adapt the network topology. Each Agent takes the current status of the Service Layer as input and has a Case Memory for future references. Here we cite the purpose of the case memory: "These agents can each have a Case Memory to support the learning of feedback control relationships and adaptive management of QoS requirements by utilizing a case-based reasoning technique for indexing, capturing and retrieving the feedback structures associated with QoS constraints."[19]

(Bordetsky, 1999) also describes adopting a Committee Model to collaborate the Agents within the Artificial Intelligence Agents. This technique will provide the necessary tool for the Multi Agent Simulation to reach an acceptable decision.

The learning process of the Collaborating Agents is done with an Artificial Neural Network. As in Figure 27, four layers of the ANN provide an hierarchical structure of determinant functions capable of learning changes in the Agents input coefficients.

There is also an integration of ANN with Case Base Reasoning Memory where the Collaborative Agent Committee decides whether the network will go into the adaptation process or into the learning process.

G. SUMMARY

In this chapter we introduced new adaptive network management architecture. The core of this new architecture is the proposal for a new Adaptive Management Protocol that will enable the operation of the new schema and the integration of an Artificial Intelligence Layer on top of a Network Management Stack that will permit the adapting the network to any tendency for instability.

[18] Bordetsky Alex, "Adaptive Management via Multiple Collaborative Agents," Chapter in book *Software Agents for Future Communication Systems*, A.L.G. Hayzelden, J. Bigham (Editors), Springer, 1999.

[19] Bordetsky Alex, "Adaptive Management via Multiple Collaborative Agents," Chapter in book *Software Agents for Future Communication Systems*, A.L.G. Hayzelden, J. Bigham (Editors), Springer, 1999.

VI. FUTURE WORK AND CONCLUSIONS

A. FUTURE WORK

This section focuses on some possible future enhancements and modifications to the architecture presented in this thesis. Many of these enhancements would add to the realistic representation of the adaptive network and provide planners of the future network centric warfare with the necessary tool to implement and realize the Information Superiority of the future.

- Implementing the Artificial Intelligence layer. This can be done in Java programming language, which is the ultimate portable language among the various operating systems and platforms.

- Interconnecting the Artificial Intelligence layer with the work done in the MANTRIP project. This will constitute the integration of the future adaptive network, capable of making its own decision for an efficient and reliable operation.

- Implementing an interface through which the administrator can monitor, intervene, and if necessary, accelerate the speed of knowledge acquisition by inducing available experience on critical or new problems.

- Creating a series of Wrapper classes for specific models of routers that would post as available libraries for the integrated system. This is because the most difficult part of the implementation is the creating of this Wrapper class as it may require over 3 KLOC. The more proprietary and capable a router is the more KLOC we need for such an implementation.

- An RFC for the new Adaptive Management Protocol that will replace the SNMP protocol, at least for military purposes and platforms. This new management protocol must endorse two-way communications with a management application and provide the necessary data to manage the network sufficiently. Such data can be bandwidth per TCP connection and the capability of creating new connections directly from the manager station.

B. CONCLUSIONS

This thesis explored the possibility of creating an adaptive network on the basis of the configuring of the SNMP protocol, which is inherent and supported by almost all of the today’s platforms and operating systems. This hypothesis was not sound as there are not enough capabilities for the SNMP protocol to support anything more than monitoring of a network. This means that the SNMP protocol can only provide the necessary data to manage the network and not for directly altering its status or tendency by modifying any of its attributes.

The next step was to circumvent this weakness by deploying wrapper classes on all the network routers. These classes can communicate with the router using the Telnet protocol and at the same time abstract this operation from the rest of the application. In such a way, the necessary instructions can be passed to any of the routers and by doing so we can intervene in the network status.

Then we adopted the technology and results provided by the MANTRIP Project in order to propose a new architecture for an adaptive network. The MANTRIP project’s architecture is directed toward a human centric network administration.

We also made a proposal for a new Adaptive Management Protocol that will provide the necessary tools for implementing the Adaptive Network Stack. Also we proposed incorporating of an Artificial Intelligence Layer in order for the new architecture to represent an adaptive network.

The Adaptive Network is the heart and core of the new network centric warfare. This thesis provided enough tools for a first implementation of what today is called the next generation network.

APPENDIX A

A. ENABLE THE SNMP SERVICE

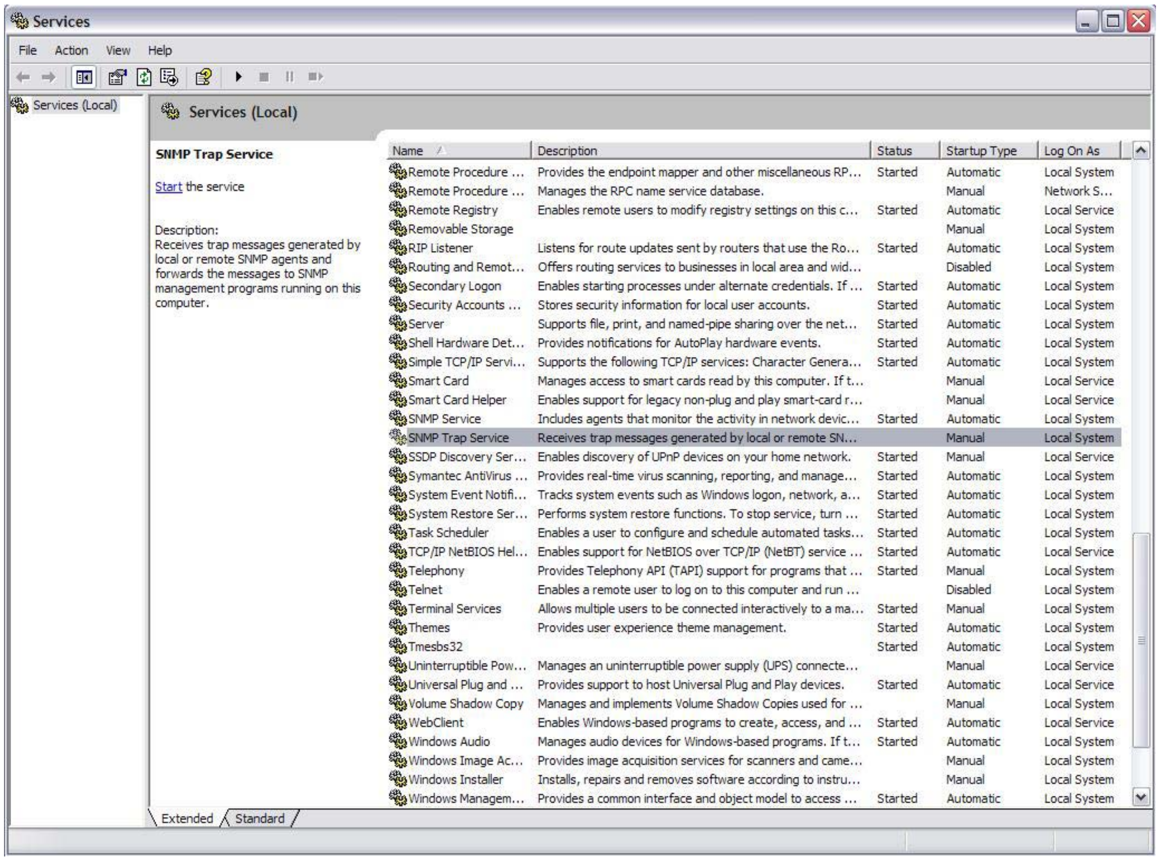
For Windows XP Professional[20]:

- Login with Administrator status.
- Open the Windows Components Wizard. To open the Windows Components wizard, click Start, point to Settings, click Control Panel, double-click Add/Remove Programs, and then click Add/Remove Windows Components.
- In Components, click Management and Monitoring Tools (but do not select or clear its check box), and then click Details.
- Select Simple Network Management Protocol check box, and click OK.
- Click Next.

To ensure that the SNMP agent is installed go to the Control Panel and click on Administrative tools, then Component Services. Highlight Services in the left frame. Then scroll through the right frame to locate the SNMP Service. However make sure that the SNMP agent is running before attempting to test it. This can be determined by looking at the status field. To ensure that the agents start up automatically right-click on SNMP Service then choose Properties. Under the General tab set the Startup type to automatic.

Next, to set the community strings for this agent, click on the Security tab. Then click on Add under the accepted Community Names. Here is the field for entering a community name. In this experiment, by disregarding the security scope, the default community string "public" will be used.

[20] Adapt from Windows 2000 help file

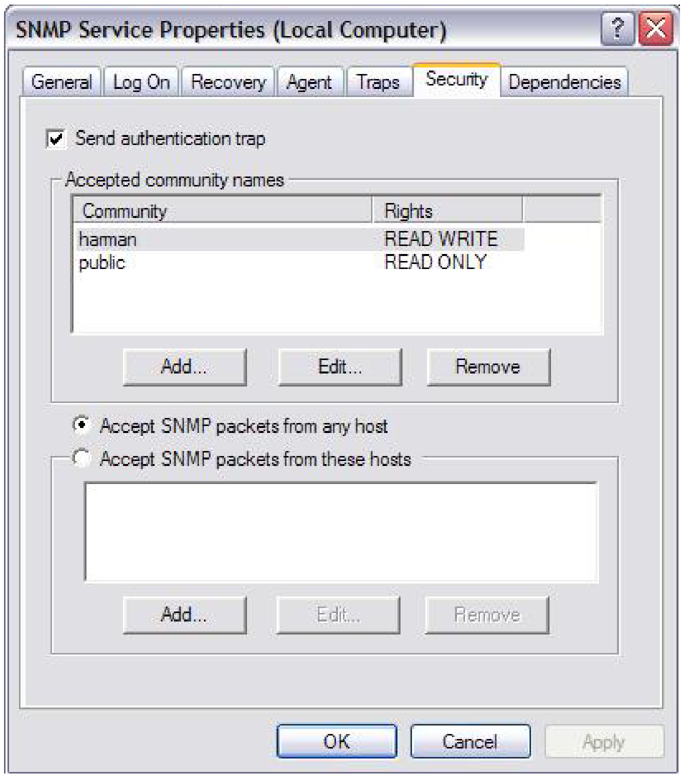


SNMP Trap Service

Start the service

Description:
Receives trap messages generated by local or remote SNMP agents and forwards the messages to SNMP management programs running on this computer.

Name	Description	Status	Startup Type	Log On As
Remote Procedure …	Provides the endpoint mapper and other miscellaneous RP…	Started	Automatic	Local System
Remote Procedure …	Manages the RPC name service database.		Manual	Network S…
Remote Registry	Enables remote users to modify registry settings on this c…	Started	Automatic	Local Service
Removable Storage			Manual	Local System
RIP Listener	Listens for route updates sent by routers that use the Ro…	Started	Automatic	Local System
Routing and Remot…	Offers routing services to businesses in local area and wid…		Disabled	Local System
Secondary Logon	Enables starting processes under alternate credentials. If …	Started	Automatic	Local System
Security Accounts …	Stores security information for local user accounts.	Started	Automatic	Local System
Server	Supports file, print, and named-pipe sharing over the net…	Started	Automatic	Local System
Shell Hardware Det…	Provides notifications for AutoPlay hardware events.	Started	Automatic	Local System
Simple TCP/IP Servi…	Supports the following TCP/IP services: Character Genera…	Started	Automatic	Local System
Smart Card	Manages access to smart cards read by this computer. If t…		Manual	Local Service
Smart Card Helper	Enables support for legacy non-plug and play smart-card r…		Manual	Local Service
SNMP Service	Includes agents that monitor the activity in network devic…	Started	Automatic	Local System
SNMP Trap Service	Receives trap messages generated by local or remote SN…		Manual	Local System
SSDP Discovery Ser…	Enables discovery of UPnP devices on your home network.	Started	Manual	Local Service
Symantec AntiVirus …	Provides real-time virus scanning, reporting, and manage…	Started	Automatic	Local System
System Event Notifi…	Tracks system events such as Windows logon, network, a…	Started	Automatic	Local System
System Restore Ser…	Performs system restore functions. To stop service, turn …	Started	Automatic	Local System
Task Scheduler	Enables a user to configure and schedule automated tasks…	Started	Automatic	Local System
TCP/IP NetBIOS Hel…	Enables support for NetBIOS over TCP/IP (NetBT) service …	Started	Automatic	Local Service
Telephony	Provides Telephony API (TAPI) support for programs that …	Started	Manual	Local System
Telnet	Enables a remote user to log on to this computer and run …		Disabled	Local System
Terminal Services	Allows multiple users to be connected interactively to a ma…	Started	Manual	Local System
Themes	Provides user experience theme management.	Started	Automatic	Local System
Tmesbs32		Started	Automatic	Local System
Uninterruptible Pow…	Manages an uninterruptible power supply (UPS) connecte…		Manual	Local Service
Universal Plug and …	Provides support to host Universal Plug and Play devices.	Started	Automatic	Local Service
Volume Shadow Copy	Manages and implements Volume Shadow Copies used for …		Manual	Local System
WebClient	Enables Windows-based programs to create, access, and …	Started	Automatic	Local Service
Windows Audio	Manages audio devices for Windows-based programs. If t…	Started	Automatic	Local System
Windows Image Ac…	Provides image acquisition services for scanners and came…		Manual	Local System
Windows Installer	Installs, repairs and removes software according to instru…		Manual	Local System
Windows Managem…	Provides a common interface and object model to access …	Started	Automatic	Local System

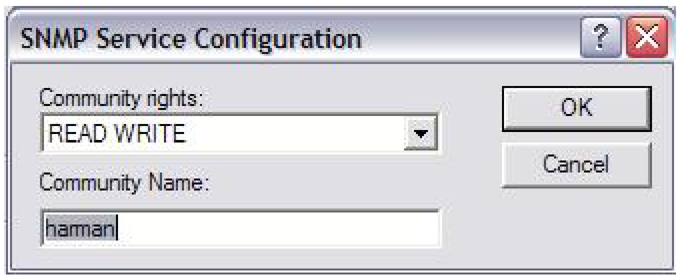


[✓] Send authentication trap

Accepted community names

Community	Rights
harman	READ WRITE
public	READ ONLY

(●) Accept SNMP packets from any host
( ) Accept SNMP packets from these hosts


SNMP Service Configuration

Community rights: READ WRITE

Community Name: harman

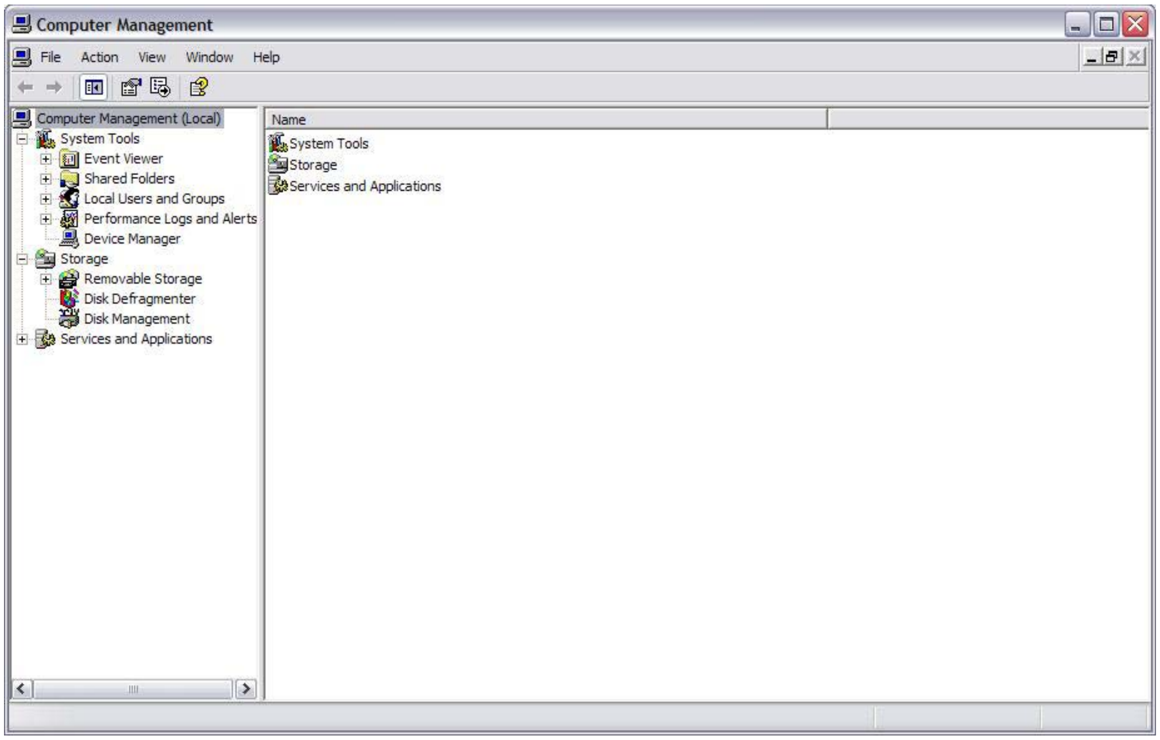
Appendix A: Figure 1. SNMP Configuration in Windows XP.

B. SNMP CONFIGURATION

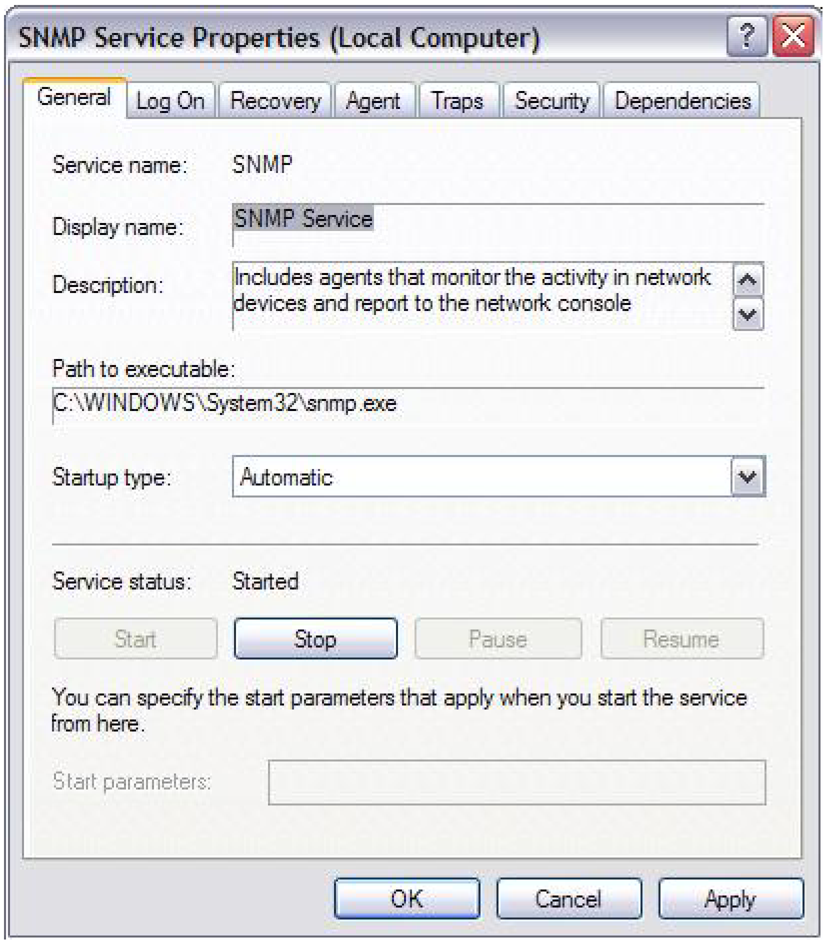
To configure agent properties for Windows XP Professional[21]:

- Open Computer Management. [Figure 2].
- In the console tree, click Services.
- In the details pane, click SNMP Service.
- On the Action menu, click Properties. [Figure 3].
- On the Agent tab, in Contact, type the name of the user or administrator for this computer. [Figure 4].
- In Location, type the physical location of the computer or the contact.
- Under Service, select the appropriate check boxes for this computer, and then click OK.

[21] Adapt from Windows 2000 help file



Appendix A: Figure 2. The Computer Management Console.



Service name: SNMP

Display name: SNMP Service

Description: Includes agents that monitor the activity in network devices and report to the network console

Path to executable:
C:\WINDOWS\System32\snmp.exe

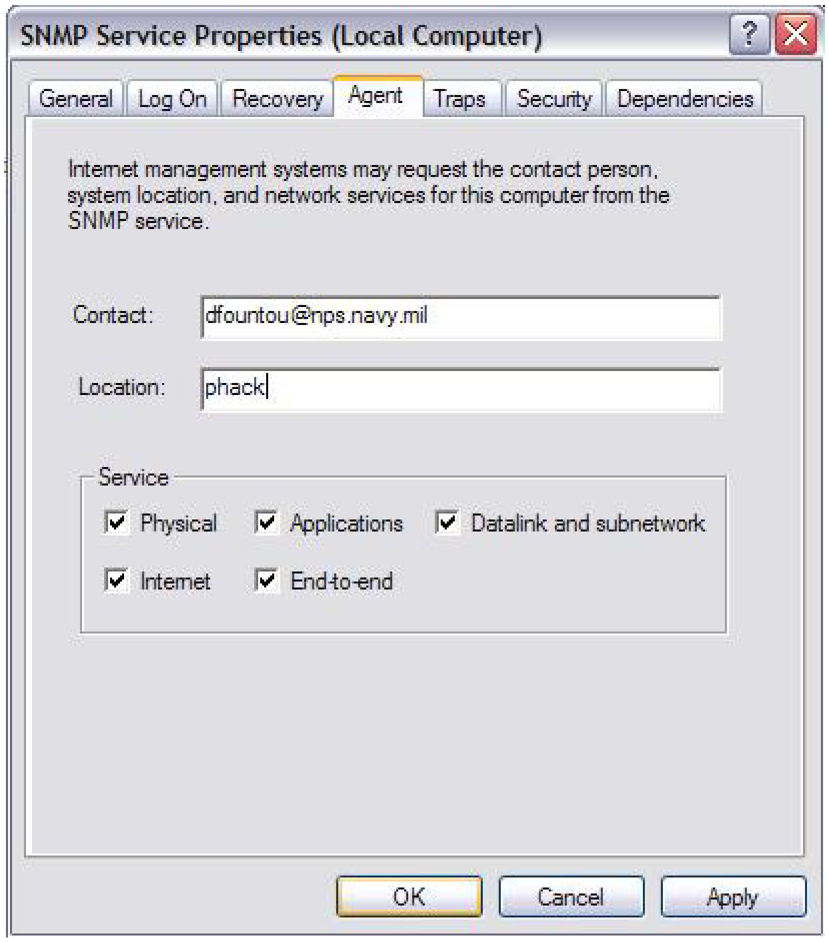
Startup type: Automatic

Service status: Started

You can specify the start parameters that apply when you start the service from here.

Start parameters:

Appendix A: Figure 3. SNMP Service Properties.



SNMP Service Properties (Local Computer)

Internet management systems may request the contact person, system location, and network services for this computer from the SNMP service.

Contact: dfountou@nps.navy.mil

Location: phack

Service

[✓] Physical [✓] Applications [✓] Datalink and subnetwork

[✓] Internet [✓] End-to-end

Appendix A: Figure 4. SNMP Service Properties: Agent Configuration.

Notes

- To open Computer Management, click Start, point to Settings, and click Control Panel. Double-click Administrative Tools and then double-click Computer Management.

- If you change the existing SNMP settings, your changes take effect immediately. If you are configuring SNMP for the first time, you must restart SNMP before these settings take effect.

APPENDIX B

A. JAVA CODE USED FOR SNMP EVALUATION

```
package snmpmanage;

/**
 * <p>Title: SNMP exploration using Inteligent Agent</p>
 * <p>Description: Use of Artificial Intelligence Agent to explore</p>
 * <p>             the possible improvement in Network Management</p>
 * <p>Copyright: Copyright (c) 2004</p>
 * <p>Company: Naval Postgraduate School</p>
 * @author Dimitrios Fountoukidis
 * @version 1.0
 */

import java.net.*;
import java.util.*;
import java.io.*;

import com.adventnet.snmp.beans.*;

public class Poller {

  String host = null;
  String community;
  SnmpTarget t;

  public Poller(String hostAddress) {

    host = hostAddress;
  }

  public void pollValue(boolean set) {
    try {

      System.out.println("Start of Polling");

      Vector columns;

      SnmpTarget target = new SnmpTarget(); //new SnmpPoller;
      target.loadMibs("C:/AdventNet/SNMPAPI/mibs/RFC1213-MIB");
      target.setWriteCommunity("harman");
      target.setTargetHost(host); // set host, or other parameters
      target.setObjectID("1.5.0");

      if (set) target.snmpSet("this is a test");
      String result = target.snmpGet();

      if (result == null) {
        System.err.println("Failed: " + target.getErrorString());
      }

      else {
        System.out.println("Response: " + result);
      }
      System.out.println("End of Polling");
      //System.exit(0);
      target.releaseResources();

    }
    catch (Exception e) {
      System.out.println(e.toString());
    }
  }

  public void resetSysName() {
    DataInputStream is = null;
    try {
      Runtime r = Runtime.getRuntime();
      try {

        Process helpProcess = r.exec("reg delete
hklm\\System\\CurrentControlSet\\Services\\SNMP\\Parameters\\RFC1156Age
nt /v sysName /f");
        if (helpProcess == null) {

          System.out.println("Couldn't reset the registry");

        }

        is = new DataInputStream(helpProcess.getInputStream());

      }
      catch (IOException ioe) {

        System.err.println("IOException: " + ioe);

      }

      String responseLine;

      while ( (responseLine = is.readLine()) != null) {
        System.out.println("Server: " + responseLine);
      }

      is.close();

    }
    catch (IOException e) {

      System.err.println("IOException: " + e);

    }

  }

  public static void main(String[] args) {
    String hostAddress;
    InetAddress localAddress = null;

    try {

      //get the local host
      localAddress = InetAddress.getLocalHost();

    }

    catch (UnknownHostException uhe) {

      System.out.println("Networking Error - Unable to resolve
localhost");

    }

    hostAddress = localAddress.getHostAddress();

    System.out.println("Host Address is : " + hostAddress);

    Poller pd = new Poller(hostAddress);
    pd.pollValue(false);
/*
    pd.resetSysName();

    try {

      System.err.println("Waiting for registry to reset...");
      Thread.sleep(40000);

    }

    catch (InterruptedException ex) {

      System.err.println("Interrupted Exception " + ex);
    }

    pd.pollValue(false);
*/
    System.exit(0);
  }
}
```

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Chairman, Code IS
   Information Sciences Department
   Naval Postgraduate School
   Monterey, California

4. Chairman, Code MV
   Modeling Virtual Environment and Simulation Department
   Naval Postgraduate School
   Monterey, California

5. Dr. Alex Bordetsky, Code 032
   Information Technology Management Department
   Naval Postgraduate School
   Monterey, California

6. John Hiles, Code MV
   Modeling Virtual Environment and Simulation Academic Committee
   Naval Postgraduate School
   Monterey, California

7. Hellenic Navy General Staff
   B2-III Department
   Mesogeion
   Cholargos, Greece

8. Hellenic Naval Academy
   Library Department
   Hadjikyriakou Avenue,
   Piraeus, 185 39
   Greece

9. Dimitrios Fountoukidis
   Lakonias 7a
   Chalandri, 152 34
   Greece

10. Panagiotis Fountoukidis
Amastridos 1
Kalamaria, 55131
Greece