Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2009)
"Energy and Technology for the Americas: Education, Innovation, Technology and Practice"
June 2-5, 2009, San Cristóbal, Venezuela.

SNMP JManager: An Open Source Didactic Application for Teaching and Learning SNMP v1/2c/3 with Support for IPv4 and IPv6

Gustavo Ayala
Universidad Central de Venezuela, Caracas, Venezuela, gayala@dearmas.com.

Pablo Poskal
Universidad Central de Venezuela, Caracas, Venezuela, pablo.poskal@ucv.ve.

Eric Gamess
Universidad Central de Venezuela, Caracas, Venezuela, egamess@kuaimare.ciens.ucv.ve.

ABSTRACT

In this paper we present *SNMP JManager*, a free open source application developed under the GNU General Public License in Java. Its main goal is to be used as a didactic application in network advanced courses at *Universidad Central de Venezuela* to support the teaching and learning of SNMP v1/2c/3. *SNMP JManager* is platform independent, easy to use, and has a user-friendly interface. Additionally, *SNMP JManager* can be a powerful application for network administrators and allows them to collect and modify the configuration of network devices through the use of SNMP v1/2c/3, import MIBs, manage traps, etc. It was developed to support both IPv4 and IPv6, which makes it one of the strongest free open source SNMP managers.

Keywords: SNMP, Didactic Applications, Open Source, IPv6, Network Management.

1. INTRODUCTION

Networks have become widely used all over the world. Their complexity has grown with the development of new devices, the implementation of emergent protocols, and the deployment of the Internet. For these reasons, network management has become an important issue. Similarly to other countries, Venezuela is facing the lack of network specialists. In the undergraduate program of Computer Sciences of *Universidad Central de Venezuela* (in English: Central University of Venezuela), many courses have been upgraded or added to the curriculum to face the problem. *Network and System Management* (in Spanish: Administración de Redes y Sistemas) is a very popular course in our University and some of its objectives include the understanding of network management standards such as SNMP.

To support the teaching and learning process, several SNMP based tools have been tried during the last few years in this course, but none has satisfied the needs. Limitations included: difficulties in installation, usability, IPv6 support, SNMP version support, source code availability, documentation, and commercial software that the University can not afford. Therefore, we decided to develop a new network management tool (*SNMP JManager*) from scratch, without the previous limitations, that presents valuable information to users. Its main goal is to be used as a teaching and learning tool for SNMP v1/2c/3 in advanced courses related to network administration at *Universidad Central de Venezuela*.

The rest of this paper is organized as follows: SNMP standard is discussed in Section 2. The new Internet Protocol (IPv6) is presented in Section 3. Related works are viewed in Section 4. *SNMP JManager* is presented and justified in Section 5. Conclusion and future work are discussed in Section 6.

2. SIMPLE NETWORK MANAGEMENT PROTOCOL

SNMP (Simple Network Management Protocol) (Case et al., 1990) (Miller, 1999) (Mauro and Schmidt, 2005) is a protocol for network management defined by the IETF (Internet Engineering Task Force) that is widely used since it is simple and easy to implement. SNMP is an application layer protocol that facilitates the exchange of management information between managed devices (agents) and NMSs (Network-Management Systems). NMSs are also called managers. It is part of the TCP/IP protocol suite and uses UDP (User Datagram Protocol) as a transport protocol. Agents listen to queries on UDP port 161 while NMSs received traps on UDP port 162.

SNMPv1 specifies five core PDUs (Protocol Data Units): *GetRequest*, *GetNextRequest*, *SetRequest*, *GetResponse*, and *Trap*. *GetRequest* is sent by a manager to retrieve the value of some objects managed by an agent. *GetNextRequest* is used iteratively by a manager to get tables, from administrated systems, such as the ARP (Address Resolution Protocol) cache, or the routing table. *SetRequest* is used by a manager to modify an object in a managed device. *GetResponse* is sent by agents to respond with data to get (*GetRequest, GetNextRequest*) and set (*SetRequest*) requests. *Trap* is used by agents to report an alert or other asynchronous events. SNMPv1 does not allow manager-to-manager interactions.

SNMPv2c (Case et al., 1996) is a revised version of SNMPv1 and includes improvements in the areas of performance, manager-to-manager communications, and error-handling. Three new PDUs were added in SNMPv2c: *GetBulkRequest*, *InformRequest*, and *Report*. The purpose of *GetBulkRequest* is to request the transfer of a potentially large amount of data including, but not limited to, the efficient and rapid retrieval of large tables. Compared to *GetNextRequest*, *GetBulkRequest* minimizes the number of requests and responses necessary to realize the transfer. *InformRequest* is sent by a manager to provide management information to a remote manager. Usage and precise semantics of *Report* are not specified; therefore, any SNMP administrative framework making use of this PDU must define it. The SNMPv2c improved error-handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1.

SNMPv1 and SNMPv2c have been criticized for their poor security model. Authentication services are performed by sending a cleartext password, known as “community string”. The use of cleartext is inherently insecure. It allows an eavesdropper to run a packet analyzer (sniffer), learn the SNMP community string and become an administrator; that is, the eavesdropper can perform any action permitted by SNMP. SNMPv3 brings significant security features by implementing a User Security Model (USM), which provides integrity, authentication and confidentiality. Integrity ensures that a packet has not been tampered with during transmission. Authentication verifies that the message is from a valid source. Confidentiality is provided through the use of encryption, and it prevents snooping by an unauthorized source. Data integrity and authentication are realized by computing message authentication codes with either MD5 (Message Digest 5) (Rivest, 1992) or SHA (Secure Hash Algorithm) (Eastlake and Jones, 2001). SNMPv3 uses DES (Data Encryption Standard) (Ferguson and Schneier, 2003) and AES (Advanced Encryption Standard) (Daemen and Rijmen, 2002) for encryption, and has different levels of security as shown in Table 1.

SMI (Structure of Management Information) (Rose and McCloghrie, 1990) (McCloghrie et al., 1999) is a framework that describes the basic types of information (*NetworkAddress*, *IpAddress*, *Counter*, *Gauge*, *TimeTicks*, etc) that can be manipulated by SNMP. A MIB (Management Information Base) is a formal description of a set of network objects that can be managed using SNMP. Standard minimal MIBs have been defined (MIB-I, MIB-II, Host Resources MIB, etc), and vendors often have private enterprise MIBs. MIB-I (McCloghrie and Rose, 1990) was defined to manage TCP/IP-based internets. MIB-II, defined in (McCloghrie and Rose, 1991), is basically an update of MIB-I. Host Resources MIB (Waldbusser and Grillo, 2000) is focused on computer management, and includes information over RAM (Random Access Memory), secondary memory, and applications that are installed in computers.

Another fundamental concept of SNMP is the notion of OIDs (Object Identifiers). An OID is a tag that allows a management entity to refer unambiguously to a particular object. OIDs are allocated in a tree fashion and

described in the MIB. The value of the OID is a sequence of integers that refers to a particular traversal of the object tree.

Table 1: Different Level of Security Provided by SNMPv3

Level	Authentication	Encryption	Features
noAuthNoPriv	No	No	Communication without authentication or encryption. This option provides no confidentiality or privacy at all, though it might be useful for certain applications such as development and debugging.
authNoPriv	MD5 or SHA	No	Communication without encryption. Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Users must be authenticated before they access any OID in the MIB of an agent.
authPriv	MD5 or SHA	DES or AES	Communication with both authentication and encryption. Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES and AES encryption in addition to authentication. Users must be authenticated before they access any OID in the MIB of an agent. In addition, all of the requests and responses between the management application and the managed entity are encrypted, so that all the data is secure.

3. INTERNET PROTOCOL VERSION 6

Due to its rapid and unexpected growth, Internet has been facing serious problems in the last few years. Lack of adequate IPv4 address space may be slowing down the development of Internet and new applications. Several proposals have been developed and implemented to solve these problems. A popular solution is NAT (Network Address Translation) that consists of hiding networks with private IPv4 addresses behind a NAT-enabled router with few public IPv4 addresses. As traffic passes from the private networks to Internet (outgoing packets), the source address in each packet is translated on the fly from the private addresses to the public addresses by the NAT-enabled router. Similarly, the destination address in each packet for incoming traffic is translated by the NAT-enabled router. However, NAT is a partial solution since it has drawbacks. Hosts behind a NAT-enabled router do not have true end-to-end connectivity and cannot participate in some Internet protocols. Services that require the initiation of connections from the Internet can be disrupted.

Another solution to the problem of the shortage of public IPv4 addresses that faces Internet consists to migrate to the new version of the Internet protocol (IPv6) (Blanchet, 2006) (Davies, 2008) (Deering and Hinden, 1998), or the coexistence between both protocols (IPv4 and IPv6). IPv6 fixes a number of problems in IPv4, such as the limited number of available IPv4 addresses. IPv6 has 128-bit addresses while IPv4 has 32-bit addresses. IPv6 also adds many improvements to IPv4 in areas such as routing and network auto-configuration. Even if IPv6 has been around for more than ten years now, there is a lack of IPv6 applications. That is, most of the common applications have not been ported to the new Internet protocol, and network administration tools are not the exception.

4. RELATED WORKS

Many open source and commercial NMSs have been tried in our network advanced courses at *Universidad Central de Venezuela* without satisfaction. Here we presents some of the most popular (*Net-SNMP*, *Getif*,

SolarWinds Engineers Edition, and *iReasoning MIB Browser*) and make a comparative analysis. This study will help us to point the learning necessities. Some of the important criteria of the study are based on learning needs from students and professors and included, but are not limited to:

- *User Interaction*: installation process, usability, user-friendly interface, parameter customization.
- *Scope of the Application*: SNMP v1/2c/3 support, MIB browser, MIB compilation, trap sender and receiver, IPv6 support, platform independence, etc.
- *Open source vs. commercial software*: The price of the software is an important issue to minimize costs. For this reason, open source software is preferred over commercial software. Besides, it is important to have access to the source code to allow students to modify and improve the application as needed.

Net-SNMP[1] is one of the most famous open source projects for network management. It is a command-line application to retrieve or manipulate information from SNMP devices, either using single requests (snmpget, snmpgetnext, snmpset), or multiple requests (snmpwalk, snmptable, snmpdelta). It is easy to install, and supports all three versions of SNMP. The Unix version works with IPv4 and IPv6, but the Windows version is limited to IPv4. The main weakness of *Net-SNMP* is its lack of graphical interface.

Getif [2] is a free multi-functional Windows GUI based network application. It is an excellent SNMP tool that allows users to collect and graph information from SNMP devices. *Getif* includes the automatic retrieval of some important tables from MIB-II (interface table, ARP cache, and routing table), and allows users to compile MIBs. *Getif* also provides basic utilities such as traceroute. It has some weaknesses since it does not support IPv6, it is limited to SNMPv1, and its unique port is for Windows.

SolarWinds Engineers Edition[3] is a set of 47 management tools that includes a SNMP manager, some utilities for Cisco Systems devices, and a GUI based interface for ping, traceroute, and DNS resolution. It is limited to IPv4, its source code is not freely available due to its commercial license, and it is only available for Windows.

iReasoning MIB Browser[4] is a powerful MIB browser that is useful for engineers to manage SNMP enabled network devices and applications. It allows users to load standard and proprietary MIBs. It includes a built-in trap receiver, supports all the three versions of SNMP as well as IPv4 and IPv6. It is a commercial product and its source code is not freely available. It has been ported to different architectures; its Windows version is very complete but the Unix versions seem to be still in the development phase.

Table 2 shows a summary of the study for all the SNMP based applications. For each criterion, a grade was given between 0 and 5 stars, where 0 stars indicate a lack or a very poor support, and 5 stars represent an optimum or efficient implementation in the application. As we can observe, most of the applications have good support for SNMPv1, SNMPv2c and IPv4, are easy to install, but only a few have support for SNMPv3 and IPv6, or can be used on different platforms.

[1] http://www.net-snmp.org
[2] http://www.wtcs.org/snmp4tpc/getif.htm
[3] http://www.solarwinds.com
[4] http://tl1.ireasoning.com/mibbrowser.shtml

Table 2: Comparative Study of Different SNMP Based Applications

Category	*Net-SNMP*	*GetIf*	*SolarWinds*	*iReasoning*
Installation	★★★★★	★★★★★	★★★★★	★★★★★
Usability	★★½☆☆	★★☆☆☆	★★★★★	★★★★★
Graphical User Interface	☆☆☆☆☆	★★★☆☆	★★★★★	★★★★★
IPv4 Support	★★★★★	★★★★★	★★★★★	★★★★★
IPv6 Support	★★½☆☆	☆☆☆☆☆	☆☆☆☆☆	★★★★★
SNMPv1 Support	★★★★★	★★★★★	★★★★★	★★★★★
SNMPv2c Support	★★★★★	☆☆☆☆☆	★★★★★	★★★★★
SNMPv3 Support	★★★★★	☆☆☆☆☆	☆☆☆☆☆	★★★★★
Allows MIBs Import	★★★★★	★★★★★	★★★★★	★★★★★
User Help	★☆☆☆☆	★★★★★	★★★★★	★★★★★
Source Code Availability	★★★★★	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆
Multi-Platform Support	★★★★☆	★☆☆☆☆	★☆☆☆☆	★★☆☆☆
Final Grade	3.75 / 5	2.58 / 5	3.41 / 5	4.33 / 5

5. SNMP JMANAGER

Since we could not find an management application that satisfies the requirements for the teaching and learning of SNMP, we decided to design and develop a brand new application, called *SNMP JManager* (Ayala and Poskal, 2008), based on the following criteria: usability, user-friendly interface, easy installation process, possibility to customize parameters, SNMP v1/2c/3 support, MIB browser, MIB compilation, trap sender and receiver, IPv6 support, and open source software.

SNMP JManager is a free and open source network management application written in Java which supports SNMP v1/2c/3. It has the basic functions of the SNMP protocol such as *GetRequest*, *GetNextRequest*, *SetRequest*, *GetBulkRequest* and advanced functions such as *Walk* that uses several *GetNextRequest* to query a network entity for a tree of information. Another important functionality of *SNMP JManager* is the retrieval of tables from managed devices, by specifying an OID. So users can access the routing table, the ARP cache, or other important tables from a remote managed device in just a few steps. Standard MIBs and vendor proprietary MIBs can be compiled and the result is presented as a tree as shown in Figure 1. *SNMP JManager* also manages traps. It can send and receive traps and the results are shown graphically with detailed information.

Since it was developed in Java, it is multi-platform and can be run in any system that implements a JVM (Java Virtual Machine). At the present time, *SNMP JManager* has an interface for English and Spanish users. Switching from one language to another is made by a simple click on the correspondent flag.

Figure 1 shows the main window of *SNMP JManager*. The application has three principal tabs (*SNMPv1*, *SNMPv2c*, and *SNMPv3*) related to each version of SNMP. On the left side, a MIB browser allows users to access OIDs in an easy way. In the upper-right window, users will have information (name, type, access mode, status and description) about the selected OID. The result of the command is shown on the lower-right window. Depending on the version of SNMP and the action selected (*GetRequest*, *GetNextRequest*, *SetRequest*, etc), the corresponding parameters will be shown for users to specify them.

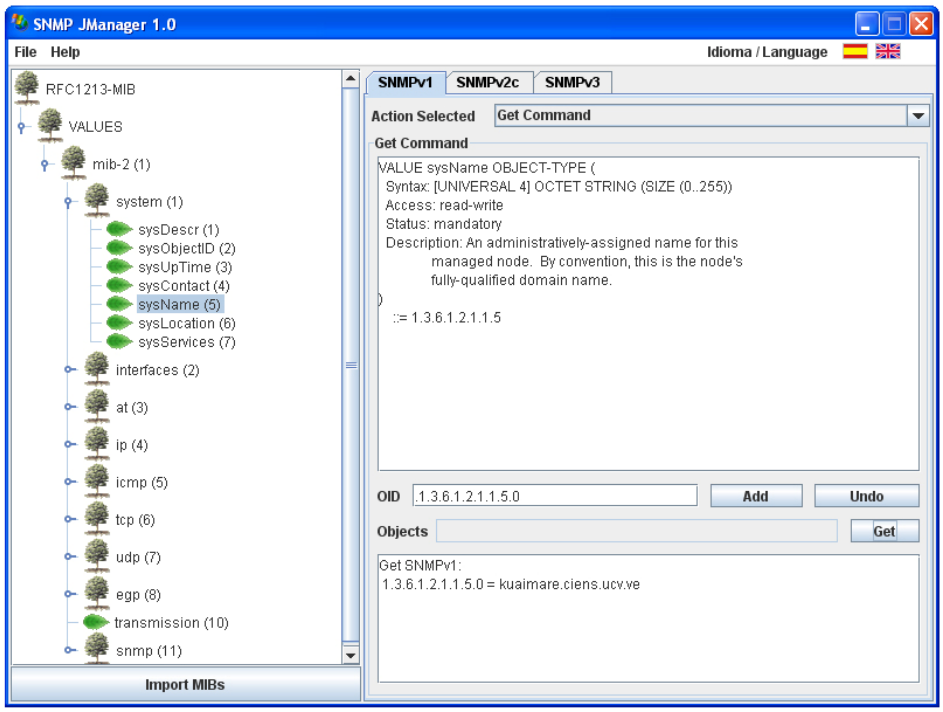

Figure 1: Main Window of *SNMP JManager*.

In Figure 2, Host Resources MIB was loaded in the MIB browser. Then, *hrSWInstalledTable* was selected and the corresponding table was retrieved (*GetTable*). This is a list of the software that is installed in the managed device, in this case a Windows computer. This table was obtained by sending several *GetNextRequest*, in a transparent way for the users.

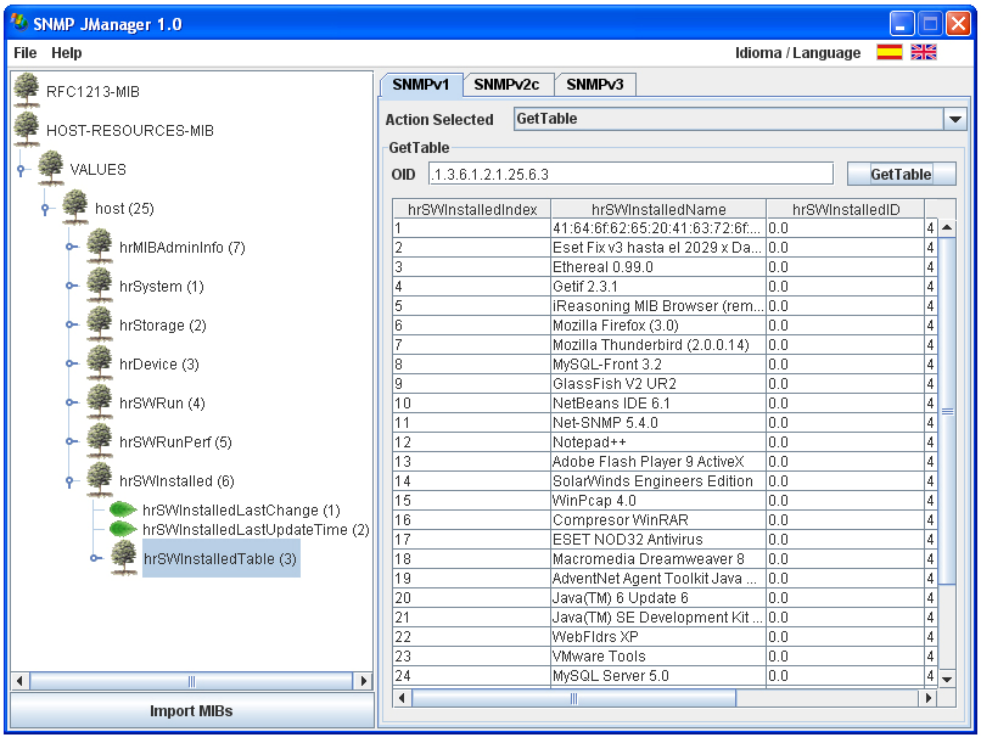

Figure 2: Retrieving of Tables.

In Figure 3, an *InformRequest* PDU has been sent using SNMPv3. The level of security was *authPriv* (authentication and privacy) so the user name, the authentication password, and the encryption password were specified. The remote manager was indicated by a link-local IPv6 address (fe80::20c:29ff:fef6:e9a5).

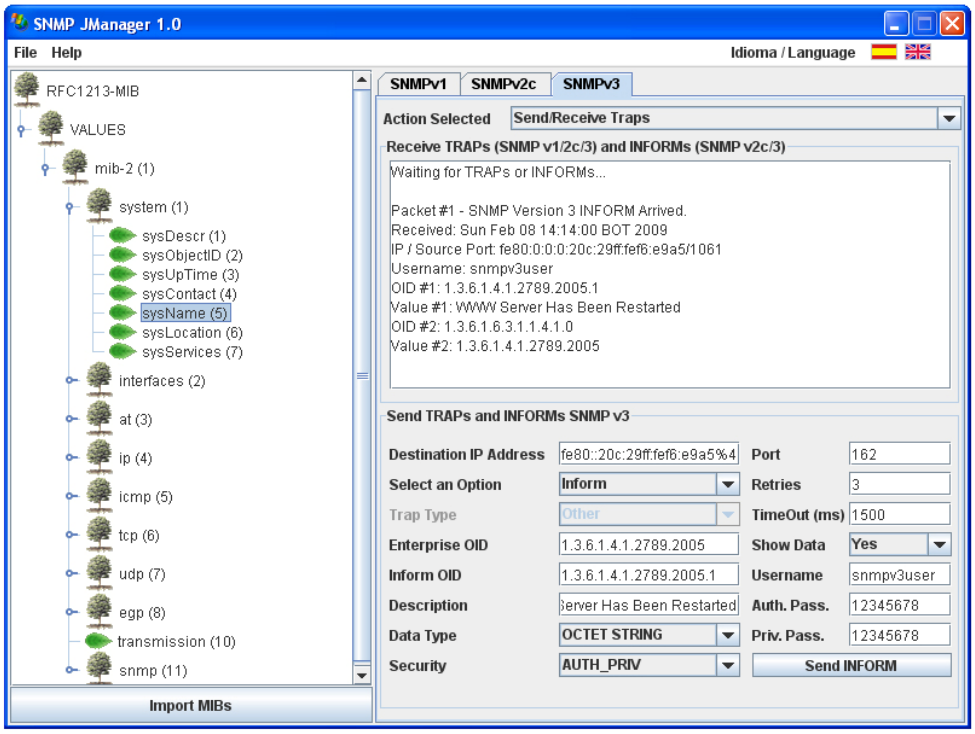

Figure 3: Sending of an InformRequest PDU.

The development of the application was based on agile programming (Martin, 2002), using XP (eXtreme Programming) method, where a series of iterations are defined: planning, designing, coding, and testing. In the planning iteration the functionalities were defined, specifying the interaction between users and the application. Here, use-case diagrams were employed to model the application functionality for every module. Then, in the designing iteration, the application structure was delimited, according to functionalities defined during the planning iteration. The code was developed in Java during the coding iteration. Finally, in the testing iteration, functionality tests were made in order to verify if the required results were obtained.

To develop the application, we used two open source packages (*SNMP4j* and *Mibble*) that extend Java functionality. *SNMP4j*[5] is a state-of-the-art SNMP implementation in Java. It allows the development of SNMP v1/2c/3 applications. *Mibble*[6] is an open-source MIB parser written in Java. It can be used to read MIB files as well as simple ASN.1 (Larmouth, 1999) (Dubuisson, 2000) files. *Mibble* simplifies the access to MIB information, such as OIDs, object types and descriptions.

Figure 4 shows a diagram of the principal classes of *SNMP JManager* and their relationships. Class *managerSNMP* (the main class of the application) has reference to *MibTreeBuilder*, *MibNode*, *recibirTrapInform*, *enviarTrapInform*, *SNMPv1*, *SNMPv2c*, *SNMPv3*, *walk* and *getTable*, which are the principal classes.

[5] http://www.snmp4j.org
[6] http://www.mibble.org

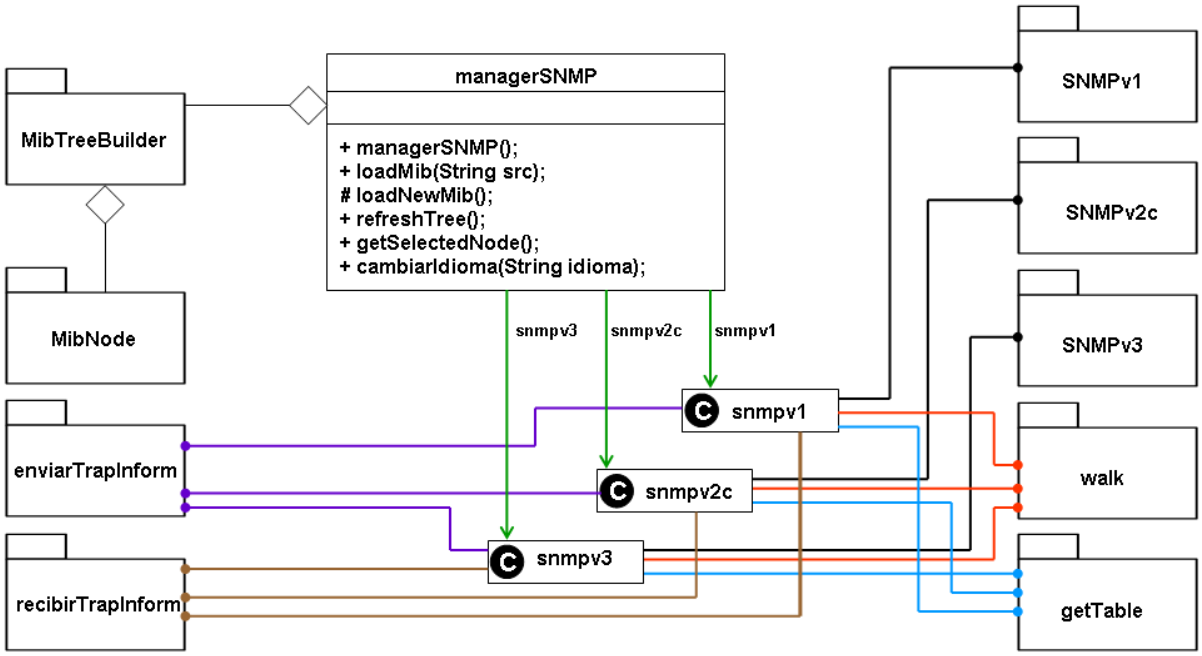

Figure 4: Diagram of the Principal Classes.

Classes *MibTreeBuilder* and *MibNode* encapsulate attributes and methods to load new MIBs into the application, to display the objects as a tree, and to get information of all the objects. Class *SNMPv1* implements all the commands of SNMPv1 which have common parameters (IP address of the agent, UDP port of the agent, read or write community, number of intents, timeout, etc). Class *SNMPv2c* is basically similar to *SNMPv1*, but includes additional functionalities for *GetBulkRequest*. Since the security model has chance from a cleartext community string sent over the network in SNMP v1/2c to the USM model of SNMPv3, the methods of class *SNMPv3* have new parameters as: user name, authentication method and key, as well as encryption method and key. Class *walk* has methods to retrieve a tree of information from remote network entities, by using several *GetNextRequest*. With class *getTable*, users can get information stored in a SNMP table of managed devices. It is very useful to get the routing table, the ARP cache, the interface table, etc. Class *recibirTrapInform* has a daemon to receive *Trap* and *InformRequest* PDUs. Class *enviarTrapInform* has functionality to send *Trap* and *InformRequest* PDUs.

As we can see, *SNMP JManager* is an excellent tool to be used in the teaching and learning process of SNMP. It covers all the functionalities and all the three versions of SNMP. It has been released recently and can be downloaded from Sourceforge at http://sourceforge.net/projects/snmpjmanager. Now, it is in use in the course of *Network and System Management* at *Universidad Central de Venezuela* to teach SNMP concepts. The feed-back that we received from students and professors is very helpful and positive.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a brand new SNMP based application distributed under the GNU General Public License. *SNMP JManager* has many features that include: usability, user-friendly interface, easy installation process, parameter customization, SNMP v1/2c/3 support, MIB browser, MIB compilation, trap sender and receiver, IPv6 support, and platform independence. It is especially suitable to enforce the teaching and learning process of all SNMP versions since it presents many valuable information to users and its source code can be easily adapted to specific requirements. Our comparative study seems to indicate that it is one of the strongest open source SNMP based applications available at the present time for teaching and learning purposes. Now it is used in our University (*Universidad Central de Venezuela*) and we have received a lot of encouragement from students and professors to pursue its development.

We plan to continue the development of *SNMP JManager*. Our objective is to extend its functionality with a grapher that will allow users to graph OIDs in real-time (for example network traffic passing through the interface of a router), a packet analyzer, and some basic administrative tools (ping, traceroute, DNS resolver, etc) to make it a powerful open source solution for network administration. We also plan to add other languages (for now, only English and Spanish are offered) to *SNMP JManager* so it can be used by more people all over the world.

REFERENCES

Ayala, G. and Poskal, P. (2008). "Herramienta para la Gestión y Administración de Redes IPv4 e IPv6 Mediante la Utilización del Protocolo SNMP". Escuela de Computación, Universidad Central de Venezuela, Caracas, Venezuela.

Black, U. (1994). "Network Management Standards: SNMP, CMIP, TMN, MIBs and Object Libraries". McGraw-Hill.

Blanchet, M. (2006). "Migrating to IPv6: A Practical Guide to Implementing IPv6 in Mobile and Fixed Networks". John Wiley and Sons. 1st edition.

Case, J., Fedor, M., Schoffstall, M., and Davin, J. (1990). "A Simple Network Management Protocol (SNMP)". RFC 1157.

Case, J., McCloghrie, K., Rose, M., and Waldbusser, S. (1996). "Introduction to Community-based SNMPv2". RFC 1901.

Daemen, J. and Rijmen, V. (2002). "The Design of Rijndael: AES - The Advanced Encryption Standard". Springer-Verlag.

Davies, J. (2008). "Understanding IPv6". Microsoft Press. 2nd edition.

Deering, S. and Hinden, R. (1998). "Internet Protocol, Version 6 (IPv6) Specification". RFC 2460.

Dubuisson, O. (2000). "ASN.1: Communication between Heterogeneous Systems". Morgan Kaufmann.

Eastlake, D. and Jones, P. (2001). "US Secure Hash Algorithm 1 (SHA-1)". RFC 3174.

Ferguson, N. and Schneier, B. (2003). "Practical Cryptography". John Wiley and Sons; 1st edition.

Larmouth, J. (1999). "ASN.1 Complete". Morgan Kaufmann.

Martin, R. (2002). "Agile Software Development, Principles, Patterns, and Practices". Prentice Hall. 1st edition.

Mauro, D. and Schmidt, K. (2005). "Essential SNMP". O'Reilly. 2nd edition.

McCloghrie, K. and Rose, M. (1990). "Management Information Base for Network Management of TCP/IP-based Internets". RFC 1156.

McCloghrie, K. and Rose, M. (1991). "Management Information Base for Network Management of TCP/IP-based Internets": MIB-II. RFC 1213.

McCloghrie, K., Perkins, D., and Schoenwaelder, J. (1999). "Structure of Management Information version 2 (SMIv2)". RFC 2578.

Miller, M. (1999). "Managing Internetworks with SNMP". John Wiley and Sons. 3rd edition.

Rivest, R. (1992). "The MD5 Message-Digest Algorithm". RFC 1321.

Rose, M. and McCloghrie, K. (1990). "Structure and Identification of Management Information for TCP/IP-based Internets". RFC 1155.

Stallings, W. (1999). "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2". Addison-Wesley. 3rd edition.

Waldbusser, S. and Grillo, P. (2000). "Host Resources MIB". RFC 2790.

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.