SIMULACIÓN DE LAS PRIMITIVAS SNMPV2 EN UN ENTORNO DE UNA RED LAN

Ángel Ibarra [1] Germán Ramos [2] Washington Medina [3]
Facultad de Ingeniería en Electricidad y Computación [1] [2] [3]
Escuela Superior Politécnica del Litoral (ESPOL) [1] [2] [3]
Campus Gustavo Galindo, Km 30.5 vía Perimetral [1] [2] [3]
Apartado 09-01-5863. Guayaquil-Ecuador [1] [2] [3]
xibarra@espol.edu.ec [1] gtramos@espol.edu.ec [2] wmedina@espol.edu.ec [3]

Resumen

Actualmente existen diferentes tipos de protocolo en la capa de aplicación que permiten al usuario realizar distintas tareas entre dos o más dispositivos dentro de un mismo sistema de telecomunicación. Uno de los protocolos utilizados para realizar la Gestión de una Red es SNMPv2 que lo utilizamos para nuestras redes en los diferentes escenarios, este procederá a gestionar mediante el uso de comunidades para la autenticación entre los distintos dispositivos. Disponemos de diversas formas de lograr simular una red, incluso algunas nos permiten tener lo más semejante a las redes físicas existentes. En los escenarios vamos a implementar las primitivas SNMPv2 en las diferentes redes LAN por cual utilizaremos el software VIRTUALBOX para simular las redes y los dispositivos, también usaremos el software Whatsup Gold para generar las primitivas SNMPv2 y el software Wireshark para observar y analizar los resultados de los escenarios. Se utilizará el modelo de gestión "gestor-agente", donde el gestor es el centro de mando que nos ayudará a organizar, planificar y controlar. El agente es el que hace las operaciones de gestión realizadas por el gestor en la red. Se diseñaron tres escenarios con una visión de crecimiento y escalabilidad, guiándonos en las proyecciones de crecimiento en las empresas, dentro de cada escenario simulamos las primitivas: "get", "set" y "trap". Con las tres primitivas obtuvimos los resultados esperados: con el "get" obteniendo los valores solicitados a los agentes, con el "set" editando los valores en los agentes y con la "trap" recibiendo alertas.

Palabras Claves: *SNMP, SNMPV2, MIB, OID, LAN, PDU, UDP.*

Abstract

Currently there are different types of protocols in the application layer to enable the user to perform various tasks between two or more devices within a telecommunication system. One of the protocols used for the management of a network is SNMPv2 we use it in our networks in various scenarios, it will proceed to manage it by using communities for the authentication between different devices. We have various ways to achieve simulate a network, that even allow us to have similar existing physical networks. In the scenarios we going to implement the SNMPv2 primitives in different LAN networks for which we will use the VIRTUALBOX software to simulate networks and devices, we will also the Whatsup Gold software to generate the SNMPv2 primitives and the Wireshark software to observe and analyze the results of the scenarios. Management model "manager - agent" will be used, in which the manager is the command center that will help us to organize, plan and control. The agent is the one that makes the management operations made by the manager in the network. Three scenarios were designed with a vision of growth and scalability guided by the projections of growth in the companies, inside of every scenario we simulate the primitives: "get", "set" and "trap". With the three primitives we obtained the expected results: with the "get" obtaining the results requested by the agents with the "set" editing the values in the agents and with the "trap" receiving alerts.

Keywords*: SNMP, SNMPV2, MIB, OID, LAN, PDU, UDP.*

1. Introducción

En el mundo de las telecomunicaciones, está en demanda disponer de redes con mayor control y administración de los dispositivos que las integran, lo cual solucionaremos realizando Gestión sobre la Red.

Con nuestro proyecto mostraremos como utilizar la simulación de las primitivas SNMPv2 en un entorno de red LAN, obteniendo resultados parecidos a los proporcionados por una red LAN con dispositivos físicos. Para lograrlo trataremos de cumplir con los siguientes objetivos:

- Describir las funciones de las primitivas de SNMPv2 para lograr una buena Gestión de redes LAN.
- Identificar el ambiente y los escenarios a utilizar para la simulación de las primitivas SNMPv2.
- Desarrollar la gestión de las primitivas SNMPv2 en la simulación de los escenarios propuestos.

Las primitivas realizarán gestión sobre los servidores ejes en cada escenario, para lo cual disponemos de MIBS propietarias y genéricas. De los objetos disponibles con las MIBS, se realizará la simulación de las primitivas GET, SET y TRAP sobre un objeto que represente a un recurso importante para el servidor.

2. Gestión de redes

La simulación de las primitivas SNMPV2 en una red LAN nos permitirá tener planificación, organización, supervisión y control de los elementos de nuestra red LAN. Garantizándonos el proporcionar un adecuado nivel de servicio, de acuerdo a un costo determinado.

En nuestro caso la topología a implementar es la de bus la cual utiliza un solo cable backbone en donde se conectarán todos los dispositivos con los que formaremos nuestros escenarios.

2.1 Modelo gestor-agente

El modelo de gestión de red que implementaremos será el gestor-agente, en el cual intervienen los siguientes elementos:

- el agente,
- el gestor,
- el protocolo con que se gestiona, y
- la base de información gestionada (MIB, Management Information Base). [2]

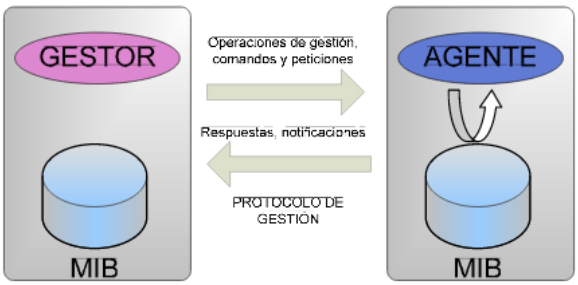

Figura 1. Protocolo de Gestión [2]

SNMP

SNMP proporciona a sus usuarios un conjunto de operaciones simples para la administración remota a dispositivos.

SNMP se comunica por medio de UDP como protocolo de transporte para pasar datos entre el gestor y los agentes.

SNMP utiliza el puerto 161 para el envío y recepción de solicitudes y el puerto 162 para la transmisión de las notificaciones de incidencias.

SNMPv2

Este protocolo de gestión nos permitirá especificar las políticas de acceso SNMP para SNMPv2. Una política de acceso SNMP es una relación administrativa que implique una relación entre una comunidad SNMP, un modo de acceso, y una vista MIB.

SNMPv2 añade y mejora algunas operaciones de protocolo. En SNMPv2 se definen las siguientes operaciones:

• GET REQUEST
• GET NEXT REQUEST
• SET REQUEST
• GET RESPONSE
• TRAP
• GET BULK
• INFORM.

Estructura del Protocolo SNMPv2

Así se muestra el mensaje de SNMPv2:

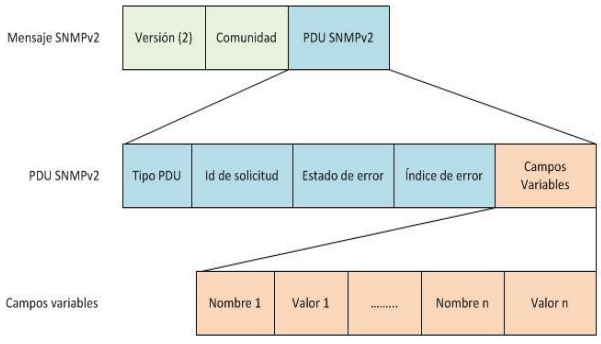

Figura 5. Mensaje SNMPv2

Para SNMPv2 la PDU de GET REQUEST, GET NEXT REQUEST, GET RESPONSE, SET REQUEST y TRAP se muestra en la siguiente figura:

Tipo PDU	Id de solicitud	Estado de error	Índice de error	Campos Variables

Figura 6. PDU SNMPv2

El formato de la PDU GET BULK se muestra en la siguiente figura:

Tipo PDU	Id de solicitud	Sin repetidores	Repeticiones Máxima	Campos Variables

Figura 7. PDU GET BULK

3. Simuladores y programas para gestionar SNMPv2 en una red LAN

3.1 Virtualbox

Figura 8. Logo del programa Virtualbox [8]

VIRTUALBOX es un poderoso programa de virtualización que corre en la arquitectura x86 y AMD64, actualmente en desarrollo por ORACLE, creado para uso empresarial y de usuarios.

VIRTUALBOX es la única solución profesional que está libremente disponible como programa de código abierto bajo los términos de la GNU General Public License (GPL) versión 2. [8]

Con VIRTUALBOX dispondremos de múltiples dispositivos agentes cuyos recursos provienen del dispositivo físico sobre el cual está instalado.

Entre los sistemas operativos existentes las máquinas virtuales manejarán Windows XP, Fedora 15 y Windows 7.

Con VIRTUALBOX lograremos cumplir con todos los objetivos planteados y también realizar una óptima demostración de la simulación de las primitivas SNMPv2 en una Red LAN.

La simulación de la red, será realizada por VIRTUALBOX mediante la selección de las subredes privadas a utilizar y asignando las direcciones IP de la subred a los equipos, con esto será suficiente para que formen parte de la subred y tener nuestra red LAN.

3.2 Whatsup gold premium edition v16

De entre los programas existentes en el mercado para Gestionar la Red, WHATSUP GOLD ha llegado a ser implementado en Centros de Datos de distintos países, siendo una solución confiable y segura que nos ofrece libertad de uso, escalable y ampliable.

WHATSUP GOLD nos permitirá realizar Gestión con SNMP de nuestra red sin muchas complicaciones permitiendo tener una completa cobertura de los dispositivos y aplicaciones que la integran.

Requisitos mínimos para la instalación de WHATSUP GOLD

Para poder proceder con la instalación del programa antes deberemos revisar que la máquina virtual cumpla con los requisitos, estos son:

- Procesador de doble núcleo.
- 2GHz de velocidad del procesador.
- Microsoft Windows Server 2003/Windows Server 2008/Windows Vista/ Windows 7
- 2GB de memoria RAM.
- 2GB mínimo de espacio en disco duro.
- 100 Mbps en la tarjeta de interfaz de red.

3.3 Wireshark

WIRESHARK es un programa libre, siendo accesible a administradores de redes, ingenieros, desarrolladores y estudiantes de redes disponible para la mayoría de los sistemas operativos.

Entre las características principales están poseer una interfaz flexible, captura de datos de la red, capaz de lograr gran filtrado, compatibilidad con varios protocolos, entre otras más.

4. Diseño de escenarios para la simulación de una red LAN por SNMPv2 con whatsup gold y máquinas virtuales

4.1 Escenarios

Teniendo ya definido que nuestros escenarios van a desarrollarse en un ambiente de empresa nos enfocaremos en aquellos que son administrados por el usuario y destinados a brindar servicios para los usuarios.

Los dispositivos dentro de la red LAN deberán estar en capacidad de brindar gran calidad de servicio y escalabilidad.

Como nuestro proyecto está orientado a desarrollarse sobre una estructura de simulación, usaremos máquinas virtuales. Al estar las máquinas virtuales sobre un programa como VIRTUALBOX los recursos de CPU, Memoria RAM, Red y Almacenamiento de la máquina virtual se reflejarán en el GESTOR mediante las primitivas del protocolo SNMPv2: GET REQUEST, GET NEXT REQUEST o GET BULK REQUEST.

En el primer escenario resolveremos uno de los primeros problemas a presentarse como lo es la necesidad de poder compartir los archivos que se manejan internamente.

El desarrollo de la red LAN consta de los elementos que son: el usuario, un servidor de archivos y el gestor. Estos trabajarán con los siguientes sistemas operativos:

- WINDOWS XP como usuario.
- FEDORA15 como servidor de archivos.
- WINDOWS 7 como gestor.

El servidor de archivos proporciona a la red el servicio SMB el cual se obtiene teniendo instalado el paquete SAMBA que es el que permite a los sistemas operativos Fedora15 poder proporcionar la administración, almacenamiento y configuración de los archivos almacenados.

Todos los equipos se manejan con el modelo TCP/IP para la comunicación entre ellos, y a la red formada se le asignara la siguiente subred: 172.16.1.0/24 dentro de la cual estarán asignadas:

- WINDOWS XP: 172.16.1.2/24
- Servidor de archivos: 172.16.1.3/24
- WINDOWS 7: 172.16.1.5/24

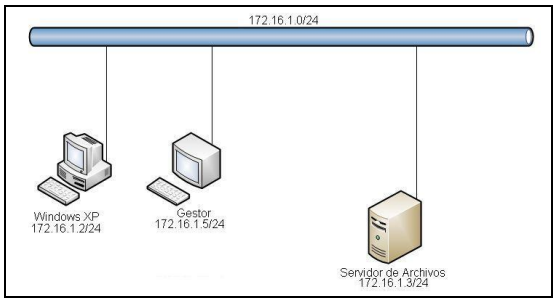

Figura 9. Diagrama Escenario 1

En el escenario 2 vamos a agregar un servidor de correos a la red LAN.

El servidor de correos nos proporciona los servicios POP y SENDMAIL los cuales ya se encuentran incluidos en el sistema operativo FEDORA 15. Nos permitirá enviar y recibir correos entre todos los usuarios que forman parte de la red LAN.

Ahora disponemos de 4 dispositivos que formarán nuestro escenario, la subred será la misma y las IPs asignadas son las siguientes:

- WINDOWS XP: 172.16.1.2/24
- Servidor de archivos: 172.16.1.3/24
- Servidor de Correos: 172.16.1.4/24
- WINDOWS 7: 172.16.1.5/24

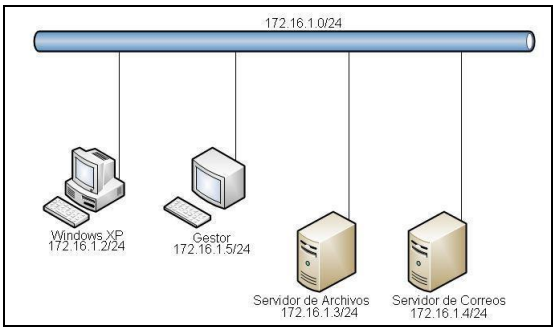

Figura 10. Diagrama Escenario 2

El escenario 3 será formado adicionando un Servidor Web (alojara las páginas web o Intranet).

La implementación de un servidor Web nos proporciona el servicio HTTPD con lo que seremos capaces de disponer de páginas web (almacenadas dentro del servidor Web), cuyo acceso será solo para la red LAN de la empresa; también se dispondrá del servicio NAMED dado por un servidor de DNS, que estará implementado dentro del Servidor Web.

Ahora disponemos de 5 dispositivos que forman parte de nuestro escenario, la subred será la misma y las IPs asignadas son las siguientes:

- WINDOWS XP: 172.16.1.2/24
- Servidor de archivos: 172.16.1.3/24
- Servidor de Correos: 172.16.1.4/24
- Servidor Web:172.16.1.1/24
- WINDOWS 7: 172.16.1.5/24

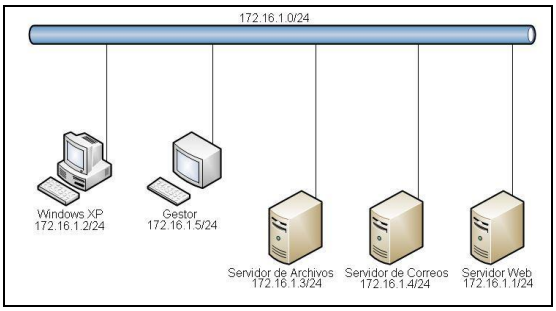

Figura 11. Diagrama Escenario 3

5. Simulación de los escenarios

En nuestros escenarios, vamos a realizar la simulación de las primitivas GET, SET y TRAP sobre cada uno de los servidores correspondiente. Para lo cual realizaremos lo siguiente:

Simulación de la primitiva get en escenario1

Realizaremos un monitoreo desde el Gestionador enviando la primitiva GET-REQUEST sobre el objeto dskUsed de OID: 1.3.6.1.4.1.2021.9.1.8.1, se creará dentro del directorio "ArchivosTesis" un archivo de 1Mb y el aumento de espacio de disco se mostrará en un gráfico 2D. Ayudándonos de Wireshark podremos realizar el análisis sobre el objeto al momento de aumentar el espacio utilizado en el disco.

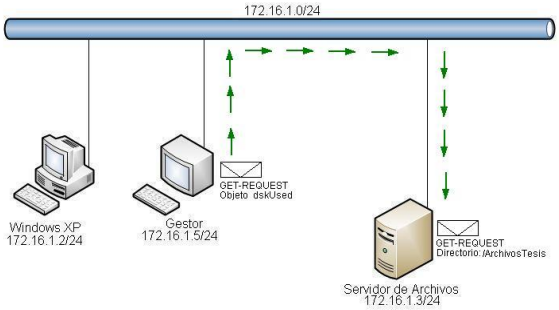

Figura 12. Simulación primitiva GET en Escenario1

Simulación de la primitiva get en escenario2

Realizaremos un monitoreo desde el Gestionador, el cual de entre todos los objetos disponibles solo realizará GET-REQUEST sobre el objeto prCount de OID: 1.3.6.1.4.1.2021.2.1.5.2, se enviará un correo electrónico para que aumente la cantidad de procesos y mostrarlo en un gráfico 2D. Ayudándonos de Wireshark podremos realizar el análisis sobre el objeto y su número de procesos.

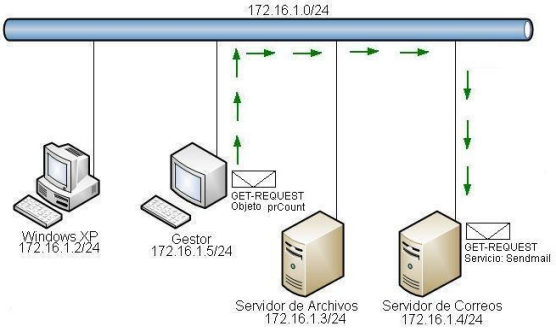

Figura 13. Simulación primitiva GET en Escenario2

Simulación de la primitiva get en escenario3

Realizaremos un monitoreo en el Gestionador lo que generará varios GET-REQUEST sobre el objeto memAvailReal de OID: 1.3.6.1.4.1.2021.4.6.0, y se irá mostrando en un gráfico 2D. Ayudándonos de Wireshark podremos realizar el análisis sobre el objeto memAvailReal.

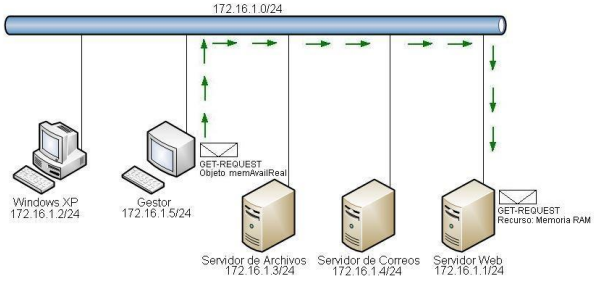

Figura 14. Simulación primitiva GET en Escenario3

Simulación de la primitiva set en escenario1

Realizaremos un cambio del valor en el objeto sysContact de OID: 1.3.6.1.2.1.1.4.0, el cual se realizará haciendo un test del SNMP Set Action que creará un SET-REQUEST al objeto sysContact, y al hacer nuevamente un Walk deberá ser visible el cambio. Ayudándonos de Wireshark podremos realizar el análisis sobre el objeto al momento del cambio.

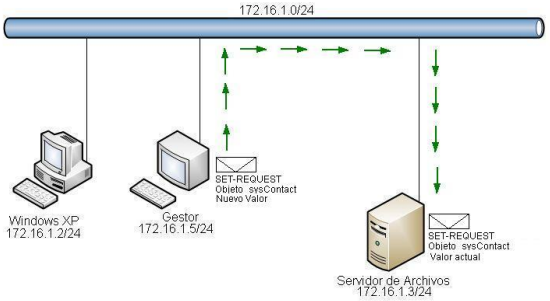

Figura 15. Simulación primitiva SET en Escenario1

Simulación de la primitiva set en escenario2

Realizaremos un cambio de valor en el objeto sysName de OID: 1.3.6.1.2.1.1.5.0, el cual al realizar un test del SNMP Set Action se enviará un SET-REQUEST al objeto sysName, al realizar nuevamente un Walk deberá ser visible el cambio. Ayudándonos de Wireshark podremos realizar el análisis sobre el objeto al momento del cambio.

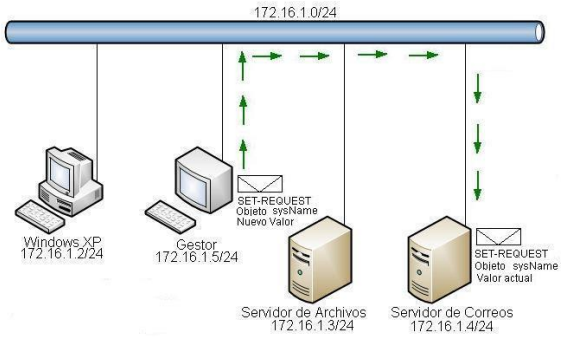

Figura 16. Simulación primitiva SET en Escenario2

Simulación de la primitiva set en escenario3

Realizaremos un cambio de valor en el objeto sysLocation de OID: 1.3.6.1.2.1.1.6.0, a lo que se realice un test del SNMP Set Action se enviará un SET-REQUEST al objeto sysLocation y con un Walk deberá ser visible el cambio. Ayudándonos de Wireshark podremos realizar el análisis sobre el objeto posterior al cambio.

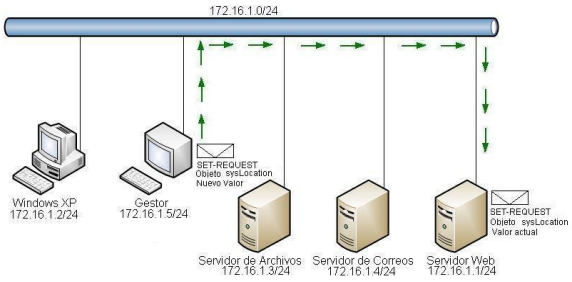

Figura 17. Simulación primitiva SET en Escenario 3

Simulación de la primitiva trap en escenario1

Vamos a proceder a detener el servicio Snmpd, esto hará que se envíe la TRAP correspondiente. Luego daremos inicio al servicio y se deberá generar otra TRAP correspondiente.

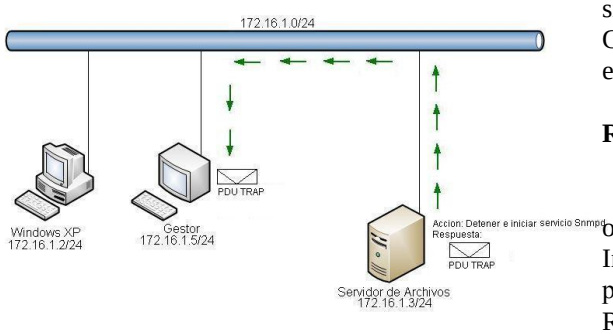

Figura 18. Simulación primitiva TRAP en Escenario 1

Simulación de la primitiva trap en escenario2

Realizaremos el cambio de la comunidad en el agente, al momento que el gestionador realice alguna primitiva GET el Servidor de Correos deberá enviar como respuesta una TRAP.

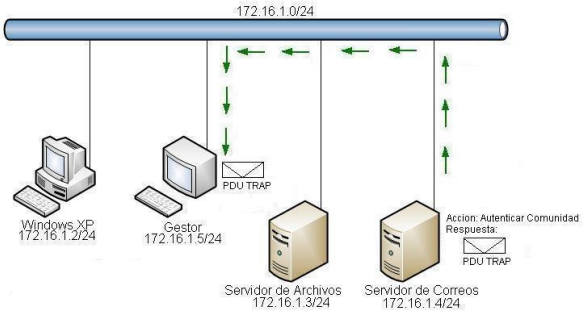

Figura 19. Simulación primitiva TRAP en Escenario 2

Simulación de la primitiva trap en escenario3

Vamos a realizar el apagado del Servidor Web y después de un tiempo prudente la encenderemos. Con lo que deberemos esperar recibir la TRAP e INFORM correspondiente tanto al evento de apagado y de encendido.

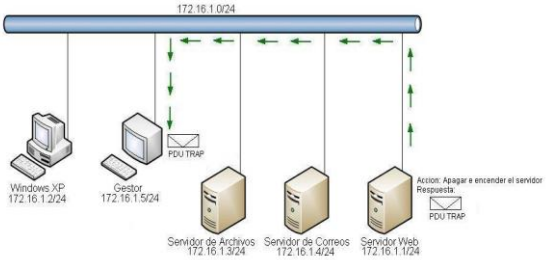

Figura 20. Simulación primitiva TRAP en Escenario 3

6. Resultados y análisis de los escenarios

Hemos podido constatar que en nuestro modelo de simulación de una Red LAN es posible realizar Gestión con un modelo Gestor-Agente y semejante a escenarios reales.

Resultados y análisis de la primitiva get

En el escenario1 la primitiva GET es realizada al objeto dskUsed cuyo valor tiene el tipo de dato Integer32. Dentro de la figura vemos que en la PDU el parámetro "data" tiene la id: 0, que se asocia a GET-REQUEST. En los campos variables (variable-bindings) vemos la OID: 1.3.6.1.4.1.2021.9.1.8.1 que está asociado al objeto dskUsed y el valor de tipo de dato Null, en espera de la respuesta, véase Figura 21.

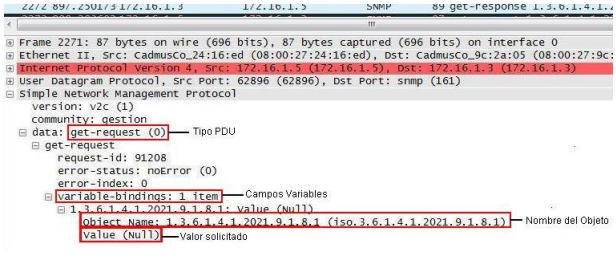

Figura 21. Análisis de PDU GET-REQUEST

Al recibir un GET-RESPONSE verificamos en su PDU el mismo request-id: 91208, lo que indica que esta es la respuesta esperada. Ahora el parámetro "data" tendra de id: 2, asociada a GET-RESPONSE. En este caso se obtuvo satisfactoriamente el valor: 7420 asociado a la OID:1.3.6.1.4.1.2021.9.1.8.1 correspondiente al objeto dskUsed, véase Figura 22.

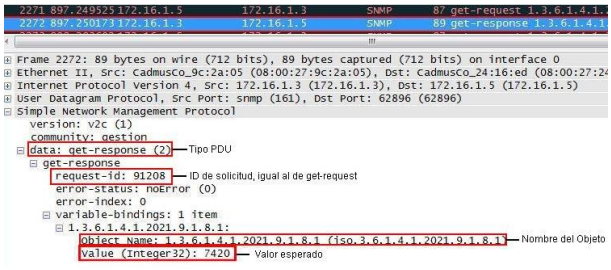

Figura 22. Análisis de PDU GET-RESPONSE

Cuando se realizó el aumento del espacio en el disco "ArchivosTesis" se comprobó que el protocolo UDP permite a SNMPv2 una rápida respuesta ante grandes cantidades de solicitudes.

En el escenario 2 vemos que la primitiva GET realizada al objeto prCount posee el tipo de dato Integer32. Dentro de los campos variables que contiene la OID: 1.3.6.1.4.1.2021.2.1.5.2 asociado a prCount y el valor de tipo de dato Null en espera de la respuesta. El comportamiento entre GET-REQUEST y su correspondiente GET-RESPONSE es similar al analizado dentro del escenario 1.

Veremos como respuesta un GET-RESPONSE, donde se obtuvo satisfactoriamente el valor: 3 equivalente al número de procesos del servicio Sendmail, asociado a la OID:1.3.6.1.4.1.2021.2.1.5.2 correspondiente al objeto prCount , véase Figura 23.

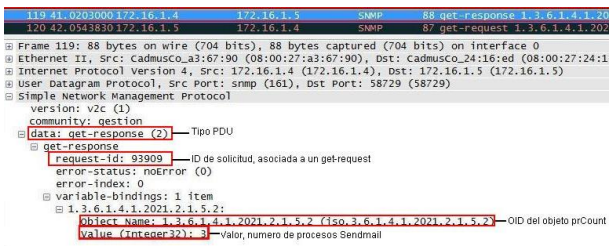

Figura 23. Análisis de PDU GET-RESPONSE Escenario 2

Cuando se realizó el envío del correo electrónico, el tiempo del que se dispone para visualizar el aumento de procesos es corto, debido a que durará lo que demore el proceso de envío de correo.

En el escenario 3 la primitiva GET se realizá al objeto memAvailReal cuyo tipo de dato es Integer32. Dentro de la PDU de GET-REQUEST veremos en los campos variables que contiene la OID: 1.3.6.1.4.1.2021.4.6.0 asociada al objeto memAvailReal y el valor de tipo de dato Null.

Comprobamos que en el PDU GET-RESPONSE se obtuvo satisfactoriamente el valor: 134840, asociado a la OID:1.3.6.1.4.1.2021.4.6.0 correspondiente al objeto memAvailReal, véase Figura 24.

Figura 24. Análisis de PDU GET-RESPONSE Escenario 3

Resultados y análisis de la primitiva set

Para el caso de las primitivas SET nos hemos ayudado de la herramienta SNMP Set Action del Gestionador, que al realizar un Test se generarán los SET-REQUEST correspondiente por cada escenario.

En el escenario 1 se realizó un SET-REQUEST al objeto sysContact cuyo valor es de tipo de dato OctetString. En la figura vemos que dentro de la PDU de SET-REQUEST el parámetro "data" se le asignó la id: 3, correspondiente a SET-REQUEST. Dentro de los campos variables se enviará el objeto, utilizando su OID: 1.3.6.1.2.1.1.4.0 correspondiente al objeto sysContact y su valor: usuario@tesis.com, debido a utilizar el tipo de dato OctetString se envía su equivalente en hexadecimal, por lo que tendrá de valor: 7573756172696f4074657369732e636f6d, véase Figura 25.

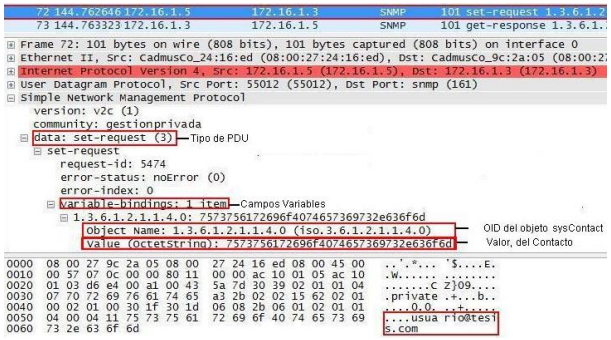

Figura 25. Análisis de PDU SET-REQUEST

Se comprueba que el cambio fue satisfactorio revisando los campos variables donde se mostrará el valor actual del objeto sysContact, véase Figura 26.

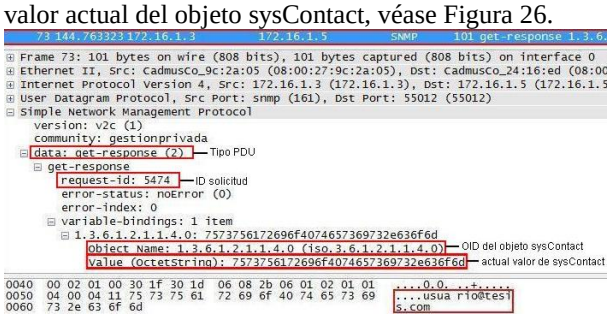

Figura 26. Análisis de PDU GET-RESPONSE de la primitiva SET

En el escenario 2 y 3 se realizará de igual manera la primitiva SET obteniendo resultados similares a los obtenidos en el escenario 1.

Resultados y análisis de la primitiva trap

En el caso de las primitivas TRAP debemos recordar que son respuestas a eventos que suceden en el dispositivo y no se relaciona a ninguna acción realizada por el Gestionador. Por lo tanto solo se dispondrá de la PDU Snmpv2-trap cuyo id: 7 para el parámetro "data" correspondiente a Snmpv2-trap enviada por el dispositivo. Es diferente a las primitivas GET y SET que trabajaban con REQUEST y RESPONSE.

También están limitadas a la cantidad de TRAPS que se dispongan en las MIBS que posea el dispositivo.

En el escenario 1 al momento de detener el servicio snmpd, el Servidor de Archivos enviará la primitiva Snmpv2-trap indicando que se detuvo el servicio, esto lo verificamos con los campos variables dando como OID: 1.3.6.1.6.3.1.1.4.1.0 del objeto snmpTrapOID y de valor la OID: 1.3.6.1.4.1.8072.4.0.2 que entre los objetos es nsNotifyShutdown (Notifica que se detuvo), véase Figura 27.

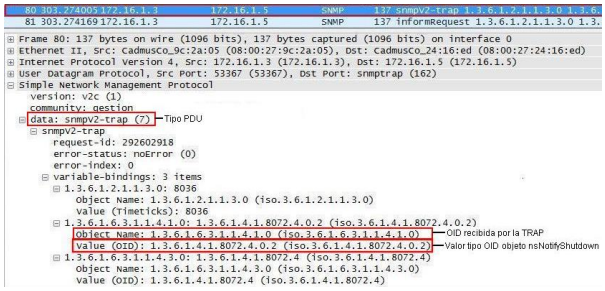

Figura 27. Análisis Snmpv2-trap al detener el servicio Snmpd

Al iniciar el servicio snmpd, obtenemos la PDU Snmpv2-trap desde el Servidor de Archivos donde observamos dentro del parámetro campos variables la OID: 1.3.6.1.6.3.1.1.4.1.0 correspondiente al objeto snmpTrapOID y de valor la OID: 1.3.6.1.6.3.1.1.5.1 que entre los objetos es llamado coldStart (Notifica que se inicio). También provee adicional la OID: 1.3.6.1.6.3.1.1.4.3.0 correspondiente al objeto snmpTrapEnterprise y de valor la OID: 1.3.6.1.4.1.8072.3.2.10 que entre los objetos es llamado linux (Núcleo o kernel de Sistema Operativo sobre el que se encuentra el agente NET-SNMP), véase Figura 28.

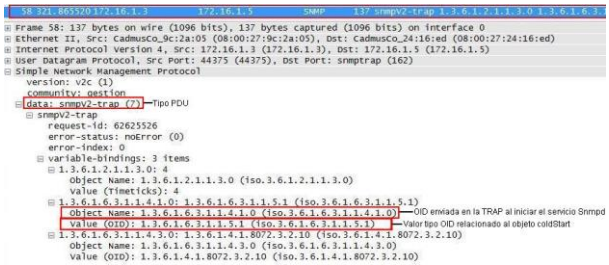

Figura 28. Análisis Snmpv2-trap al iniciar el servicio Snmpd

En el escenario 2 al momento de cambiar la comunidad en el agente, tendremos que el Servidor de Correos enviará la primitiva Snmpv2-trap al gestionador indicando que no se logra autenticar la comunidad, esto lo verificamos dentro del parámetro de campos variables dando como OID: 1.3.6.1.6.3.1.1.4.1.0 correspondiente al objeto snmpTrapOID y de valor la OID: 1.3.6.1.6.3.1.1.5.5 que entre los objetos es llamada authenticationFailure (Notifica la falla en la autenticación). También provee adicional la OID: 1.3.6.1.6.3.1.1.4.3.0 correspondiente al objeto snmpTrapEnterprise y de valor la OID: 1.3.6.1.4.1.8072.3.2, véase Figura 29.

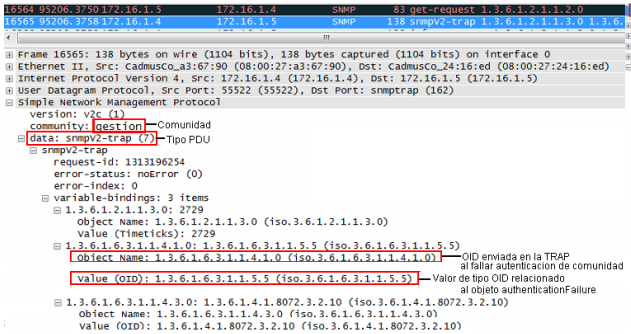

Figura 29. Análisis Snmpv2-trap al fallar autenticación de comunidad

En el escenario 3 al momento de apagar el Servidor Web se enviará la PDUSnmpv2-trap y la PDU Inform Request al gestionador indicando que se apagó el servidor. En la figura vemos la notificación de apagado del servidor, dentro de la PDU Snmpv2-trap en el parámetro de campos variables tenemos la OID: 1.3.6.1.6.3.1.1.4.1.0 correspondiente al objeto snmpTrapOID y de valor la OID: 1.3.6.1.4.1.8072.4.0.2 que entre los objetos es llamada nsNotifyShutdown. También provee adicional la OID: 1.3.6.1.6.3.1.1.4.3.0 correspondiente al objeto snmpTrapEnterprise y de valor la OID: 1.3.6.1.4.1.8072.4 que entre los objetos es llamado netSnmpNotificationPrefix (Notificación correspondiente a NET-SNMP), véase Figura 30.

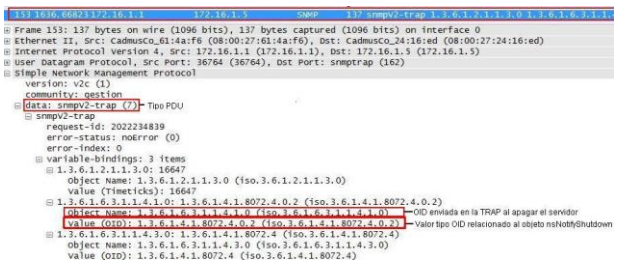

Figura 30. Análisis Snmpv2-trap al apagar el servidor.

En este escenario también se dispondrá de la PDU inform-Request cuyo id: 6 para el parámetro "data" correspondiente a inform-Request enviada por el dispositivo. Dentro de la PDU inform-Request en el parámetro de campos variables tendremos las OID y valores que se encuentran en la PDU Snmpv2-trap. El propósito de la primitiva INFORM está dado para comunicación entre gestores por motivo del evento o TRAP suscitado por lo tanto no exige mayor análisis adicional al previo.

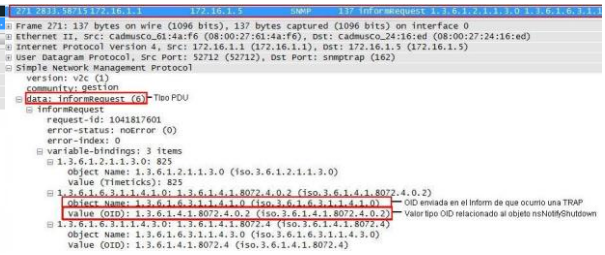

Figura 31. Análisis InformRequest al apagar el servidor.

Al encender el servidor, obtenemos la PDU Snmpv2-trap y la PDU inform-Request enviadas por el Servidor Web. Observamos dentro de la PDU Snmpv2-trap que el parámetro campos variables contiene la OID: 1.3.6.1.6.3.1.1.4.1.0 correspondiente al objeto snmpTrapOID y de valor la OID: 1.3.6.1.6.3.1.1.5.1 que entre los objetos es llamado coldStart. También provee adicional la OID: 1.3.6.1.6.3.1.1.4.3.0 correspondiente al objeto snmpTrapEnterprise y de valor la OID: 1.3.6.1.4.1.8072.3.2.10 que entre los objetos es llamado Linux, véase Figura 32.

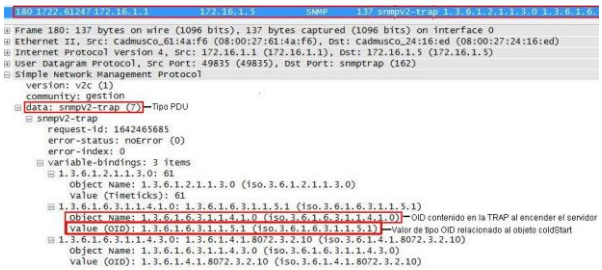

Figura 32. Análisis Snmpv2-trap al encender el servidor.

7. Conclusiones

1. La simulación de la red nos permite una mejor comprensión de cómo trabajan las primitivas SNMPv2 sin la necesidad de poseer una red real para su aprendizaje.
2. Los escenarios utilizados nos permitieron aprender cómo se crean, configuran y gestionan los servidores en cada escenario, y que forman parte de una red LAN.
3. De las primitivas SNMPv2 las primitivas GET, GET-NEXT y GET-BULK están representadas por la primitiva GET en las simulaciones debido a que realizan la misma función.
4. En el manejo del programa VIRTUALBOX la creación de la red LAN es de fácil manejo y aprendiza, siendo el más óptimo para la simulación de la red.
5. La demostración de las primitivas ayudándonos de un Gestor nos permitió aprender más sobre como maneja la información obtenida de los

objetos y las herramientas que proporciona para hacerlo.

8. Referencias

[1] Barba A. (1999), Capitulo 7 Areas funcionales de gestión, pag. 94,95

[2] Universidad Industrial de Santander, Facultad de ingenierías Fisicomecánicas http://repositorio.uis.edu.co/jspui/bitstream/123456789/8095/2/116700.pdf

[3] SNMP Agent Simulator Datasheet, http://www.webnms.com/simulator/snmp-agent-simulator-ds.html

[4] Network Management Administration Guide for Routing Devices, http://www.juniper.net/techpubs/en_US/junos/information-products/pathway-pages/network-management/network-management.pdf

[5] SNMPv3 Overview, http://www.webnms.com/net-snmp/help/technology_overview/snmnpv3_overview. html

[6] Cisco Networking Academy (s.f.), Cisco Packet Tracer, https://www.netacad.com/web/about-us/cisco-packet-tracer

[7] GNS3,http://www.gns3.net/

[8] Virtualbox, https://www.virtualbox.org/

[9] Stallings w., SNMP, SNMPV2, SNMPV3, and RMON1 and 2, 3ra. Ed., Addison- Wesley,1999

[10] Douglas R., Kevin J., Essential SNMP, 2da. Ed., O´Reilly, 2005

[11] Barba A., Gestión de red, Ediciones UPC, 1999

[12] Perkins D., Mcginnis E., Understanding SNMP MIBs, Prentice Hall,1997