[[1]](#footnote-2)

Fundamentos de Redes Digitales

Proyecto final: Monitoreo de una red LAN

Tovar Duarte, Dinorah.

Universidad Iberoamericana.

Abstract El monitoreo de una red LAN para este proyecto se ha realizado bajo los siguientes protocolos DHCP, SNMP y UDP, a su vez se virtualizo la red Lan para su máxima eficiencia en el software GNS·

# Introducción

El desarrollo de una interfaz que pueda monitorear las técnicas de una red LAN es un proceso complicado, se debe tener conocimiento de maquinas virtuales, host, routers, switches, vlans, protocolos de muestreo, programación web y sin duda conocer perfectamente un software tan amplio compo GNS3.

Se hizo uso de diferentes softwares, API’s y protocolos, los protocolos más usados fueron: DHCP, SNMP y UDP. Las API’s usadas fueron: Net-snmp, nmp de Node, Boostrap. El software más usado fue: TunTap, Gns3, Sublimetext y VirtualBox.

Todos los host simulados en la siguiente topología fueron crearon en una interfaz de Ubuntu por comodidad.

Sin embargo los routers, fueron usados con un ios de Cisco numero c3075

# Implementación

Al inicio del proceso, nos encontramos con diferentes problemas, en primera instancia se descargo GNS3 para el sistema operativo de Ubuntu, que era en si una maquina virtual generada por VirtualBox alojada en el sistema operativo de OS X El Capitan versión Beta for Developers 10.11.2, ocacionando muchos problemas debido a que se necesitaba generar diferentes host con maquinas virtuales para poder crear una pequeña topología de un red LAN.

Debido a que la maquina donde corria GNS3 era de por si una maquina virtual se encontró con el problema de que no se puede virtualizar un ambiente sobre algo ya virtualizado y por lo tanto no se pueden escalar posiciones de esa forma, sin contar que la maquina virtual principal solo contaba con 1gb de memoria RAM.

Se decidio descargar GNS3 para OS X, se instalo satisfactoriamente y de la pagina de GNS3 se decargo algunas imágenes de routers para poder trabajar, se empezó a generar un pequeña topología que contuviera dos router tres host un switch y una nueve.

Como se puede apreciar en la imagen siguiente:

Para iniciar GNS3 se inicializo con el siguiente comando en modo administrador. Debido a que desde la ultima actualización de OS X, el sistema operativo no te deja utilizar todos los archivos a los que necesita acceder GNS3

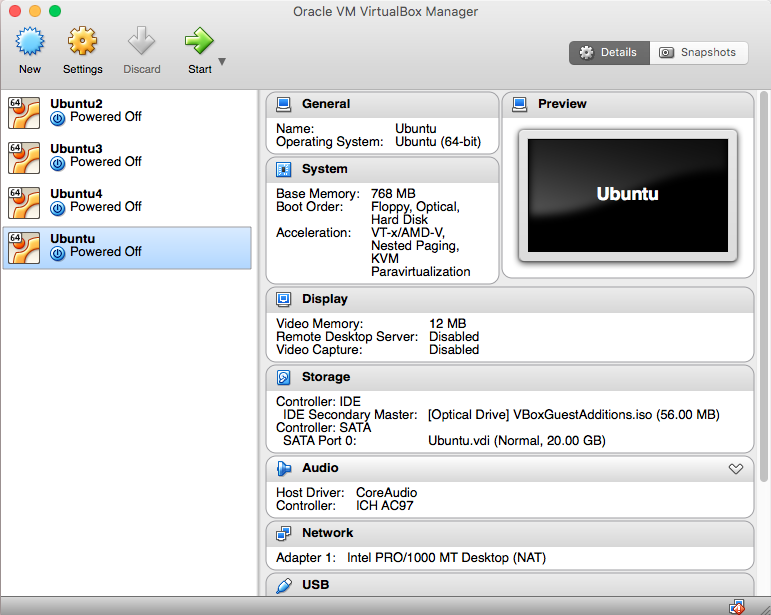
sudo /Applications/GNS3.app/Contents/MacOS/GNS3

Se decidio que debían ser pocos host por que todos los Host generados son maquinas virtuales que debían estar corriendo en el momento de las pruebas sin contar que debían tener cada instalado software diferentes, lo que lo hacían más pesadas, cada una cuenta con un IP diferente.

Host 1: con el IP: 10.0.0.3/24 255.255.255.0

Host 2: con el IP: 10.0.0.5/24 255.255.255.0

Host 3: con el IP: 10.0.0.8/24 255.255.255.0



Los Router serán configurados inicializándolos con un IDLE-PC, me encontré con un problema al no inicializarlos la primera vez asi estos generan una sobresaturación de memoria RAM.

Los routers fueron configurados en su consola privada de la siguiente manera:

Para el Router snmp:

config t

int fa0/0

ip address 10.0.0.1 255.255.255.0

no shut

end

int fa0/1

ip address 10.0.2.1 255.255.255.0

no shut

int fa0/1

ip address 10.0.2.1 255.255.255.0

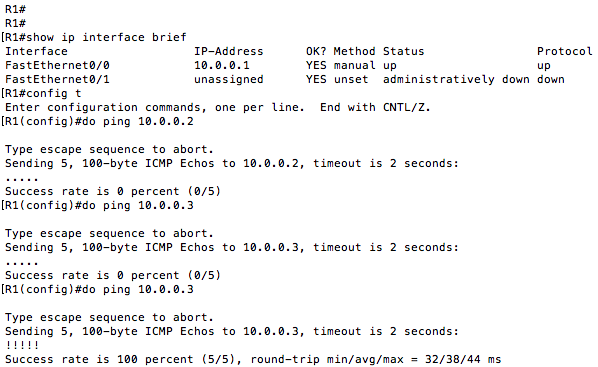
no shut

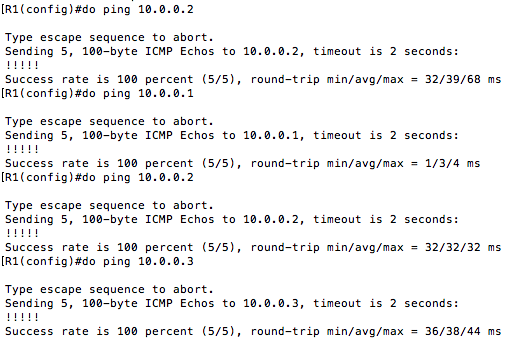
wr

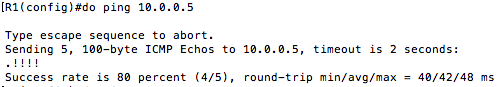
exit

snmp-server community gns3 ro

Se realizaron algunos pings a los diferentes Host y obtuvimos que se pueden comunicar







Para el otro Router:

int fa0/0

ip addr 10.0.0.6 255.255.255.0

no shut

int fa0/1

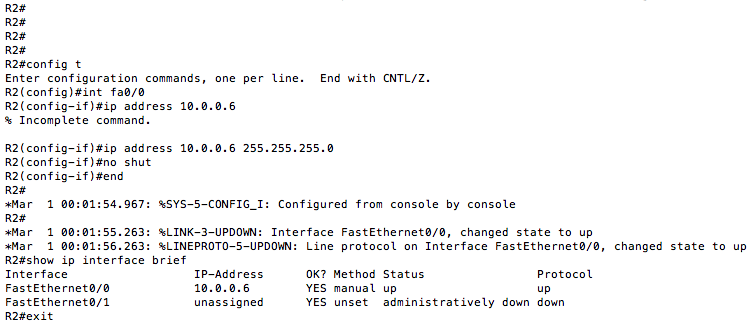
ip addr 10.0.1.1 255.255.255.0

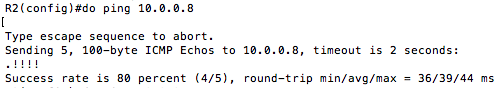
no shut

end

wr

Se realizaron algunos pings a los diferentes Host y obtuvimos que se pueden comunicar sin ningún problema





Todos los host cuentan con un Tap que genera la conexión con el router y con la Nube.

Los taps son los siguientes:

Tap0: con el IP: 10.0.0.2/24

Tap1: con el IP: 10.0.0.4/24

Tap2: con el IP: 10.0.0.7/24

Tap2: con el IP: 10.0.0.9/24

Con los siguientes comandos:

ls -l /dev | egrep 'tap|tun'

sudo chown $(id -un):$(id -gn) /dev/tap\*

sudo chmod 0755 /dev/tap\*

sudo ifconfig tap0 10.0.0.2/24 up

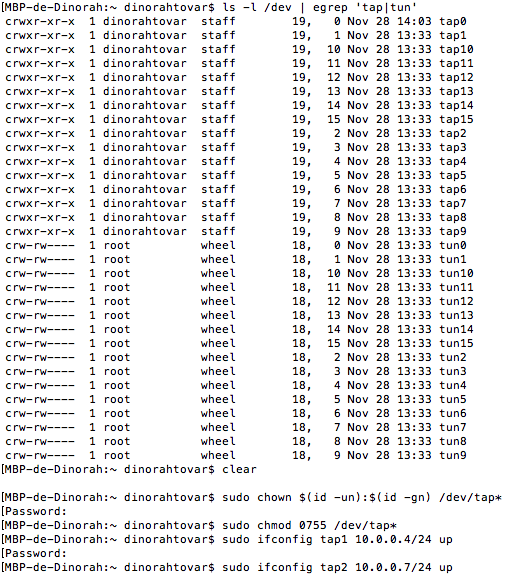
Tuntap es un pequeño pedazo de software que permite tener varios adaptadores lógicos. Puede asignar direcciones IP a estos adaptadores y generalmente los utilizan para pruebas. Por defecto, el adaptador Tuntap no puede realmente hacer algo útil por sí mismo. Se monta una serie de adaptadores virtuales llamados tun0, tun1 y así sucesivamente en el /dev del sistema de archivos. VirtualBox viene con su propio adaptador virtual llamado vboxnet pero en mi experiencia no funciona como uno lo desearía.

Vamos a tener que empezar a GNS3 como root porque Dynanips va a tratar de crear la interfaz tap0 en el equipo anfitrión. (Mac).

El comando 755 significa que el propietario ha de leer, escribir y ejecutar permisos en el archivo pero todos los demás sólo debe tener derechos para leer y ejecutar. (sin los permisos de modificación.

Ahora podemos asignarle una IP. Esto se convertirá en la dirección IP de nuestro ordenador central en el entorno virtual, así que asegúrate está en la misma subred que el router.

Esta fue la respuesta en terminal



Se verifico la configuración con el comando ifconfig Obteniendo lo siguiente:



Ahora que se pueden comunicar las maquinas desde el ambiente virtual, se tiene que verificar que las computadoras se pueden comunicar del otro lado, debido a que no se pueden mantener tantos Host ejecutándose, decidi ir haciendo cada uno de los testings por cada Host

# SNMP

Simple Network Management Protocol es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Los dispositivos que normalmente soportan SNMP incluyen routers, switches, servidores, estaciones de trabajo, impresoras, bastidores de módem y muchos más. Permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas, y planear su crecimiento.

En usos típicos SNMP, uno o más equipos administrativos, llamados gerentes, tienen la tarea de supervisión o la gestión de un grupo de hosts o dispositivos de una red informática. En cada sistema gestionado se ejecuta, en todo momento, un componente de software llamado agente que reporta la información a través de SNMP con el gerente. Los agentes SNMP exponen los datos de gestión en los sistemas administrados como variables.

El protocolo también permite realizar tareas de gestión de activos, tales como la modificación y la aplicación de una nueva configuración a través de la modificación remota de estas variables. Las variables accesibles a través de SNMP están organizadas en jerarquías. Estas jerarquías, y otros metadatos (tales como el tipo y la descripción de la variable), se describen por Bases de Información de Gestión (MIB).

GetRequest

A través de este mensaje el NMS solicita al agente retornar el valor de un objeto de interés mediante su nombre. En respuesta el agente envía una respuesta indicando el éxito o fracaso de la petición. Si la petición fue correcta, el mensaje resultante también contendrá el valor del objeto solicitado. Este mensaje puede ser usado para recoger un valor de un objeto, o varios valores de varios objetos, a través del uso de listas.

GetNextRequest

Este mensaje es usado para recorrer una tabla de objetos. Una vez que se ha usado un mensaje GetRequest para recoger el valor de un objeto, puede ser utilizado el mensaje GetNextRequest para repetir la operación con el siguiente objeto de la tabla. Siempre el resultado de la operación anterior será utilizado para la nueva consulta. De esta forma un NMS puede recorrer una tabla de longitud variable hasta que haya extraído toda la información para cada fila existente.

SetRequest

Este tipo de mensaje es utilizado por el NMS para solicitar a un agente modificar valores de objetos. Para realizar esta operación el NMS envía al agente una lista de nombres de objetos con sus correspondientes valores.

GetResponse

Este mensaje es usado por el agente para responder un mensaje GetRequest, GetNextRequest, o SetRequest. En el campo "Identificador de Request" lleva el mismo identificador que el "request" al que está respondiendo.

Trap

Una trap es generado por el agente para reportar ciertas condiciones y cambios de estado a un proceso de administración. El formato de la PDU es diferente:

Para esta practica se decidió usar la documentación de GNS3 donde nos recomiendan hacer un ping de nuestro Host a el router que administra en snmpt

ping 10.0.0.1 -c 3

snmpget -v 2c -c gns3 10.0.0.1 system.sysDescr.0

Nos regreso un String que especifica las especificaciones del router.

Eso nos confirma que el programa se esta conectando y que podemos usar snmp para poder monitorear el sistema de redes

Se hizo uso de una API de Github donde se hace uso de snmp desde Javascript pasando por Node.js

Con los siguientes comandos se puso a andar esta API:

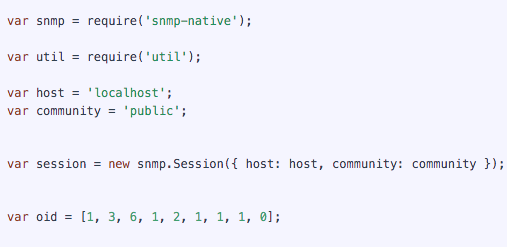
npm search

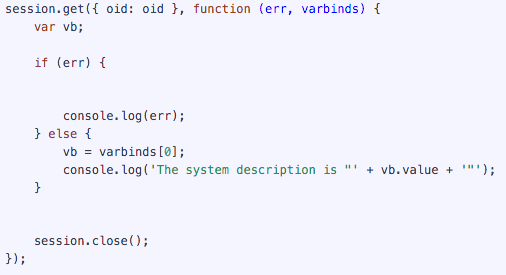
sudo npm install easyimage

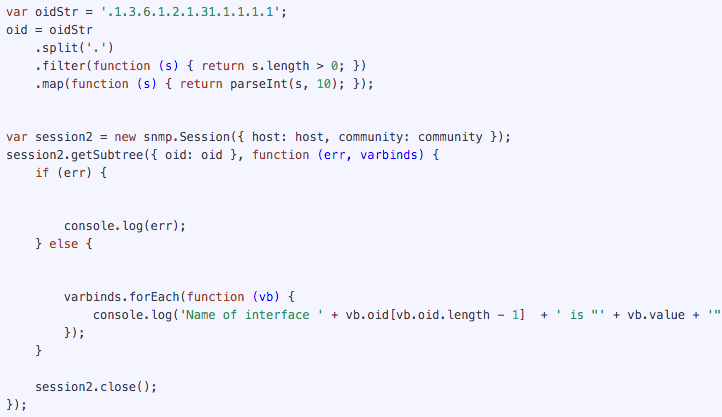
npm update

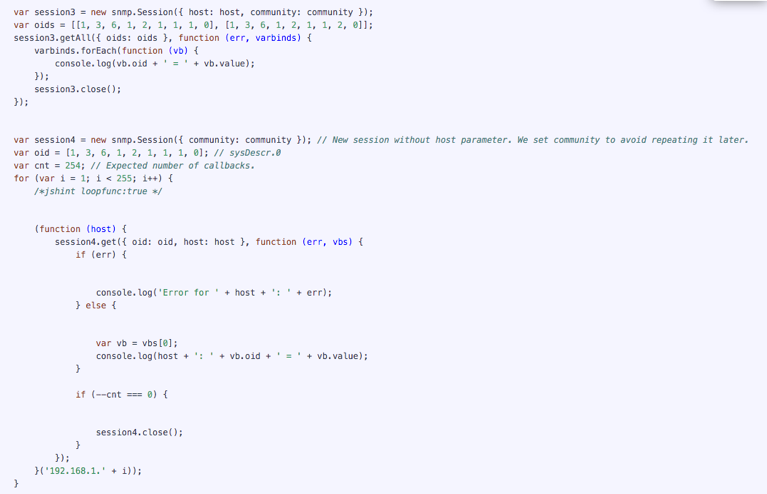
sudo npm install snmp-native

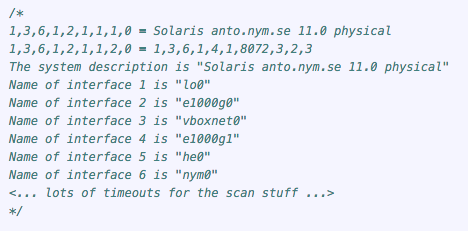
Se verifico toda la documentación para poder inicializar el programa con solo procesos en javascript











Se realizo la interfaz grafica web con ayuda del servidor XAMPP para creación de carpetas en servidores.

Despues de configurarlo se decidio hacer los principales htmls de la interfaz grafica, con ayuda de Bootstrap que es un conjunto de frameworks que te ayudan a diseñar una pagina web de manera más eficiente y rápida.

Se diseñaron 4 interfaces, la primera es un escrito documentado del proceso de la pagina, mientras que dos consecuentes son el desarrollo de tiempos de la interfaz grafica, en el primero se miden Timeticks obtenidos de los system.sysUpTime.0 referenciados a los host (En realidad a sus direcciones IP )

Se obtienen estos datos en formato de string.

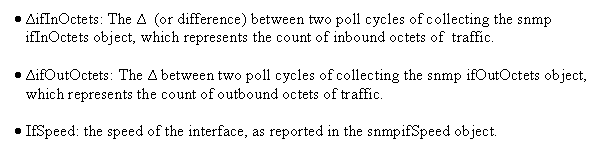
Se cortan dependiendo del IP y se alamacenan los tiempos en una variable dependiendo de sus hosts, en los que estos se ponen dependiendo del la forma de entrada.

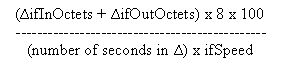
Se crean dos interfaces, una donde se puede verificar todos los datos almacenados en las variables, mientras que en otra se puede revisar el host numero 1 en un tiempo estimado de 40 segundos.

En la siguiente parte de las graficas se puede verificar el ancho de banda que los host han usado desde que se inicializo el programa, debido a la complejidad de la pagina, decidí que cada vez que sea un nuevo grupos de host guarde las variables.

Asi estás no inicialicen en 0.

Se utilizo las formulas que recomienda Cisco dependiendo de los ciclos de recolección de datos.

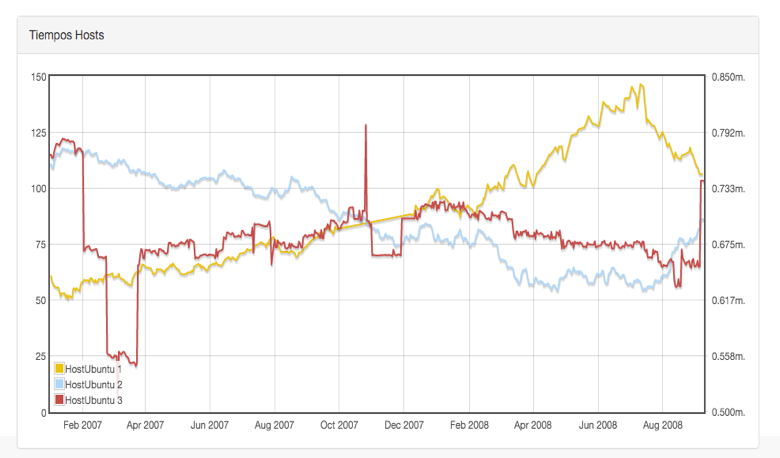


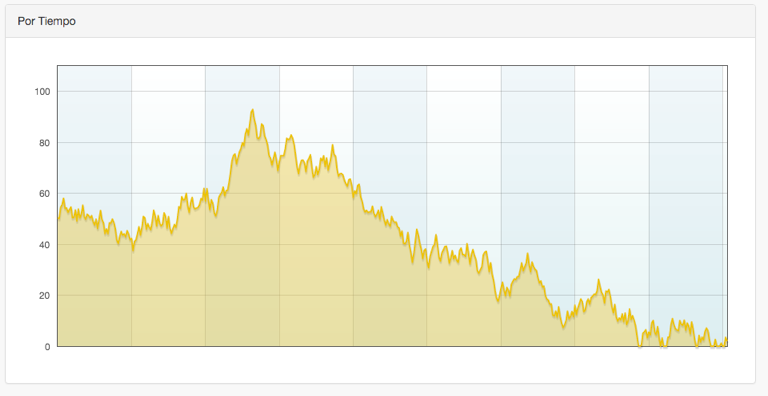


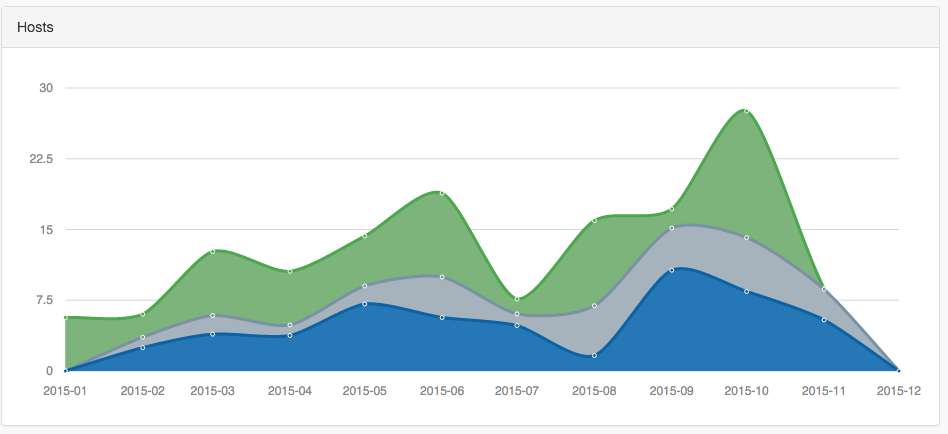
Con ayuda de distintos Jquerys que se anexan, se consiguió obtener todos los datos necesarios.

# Resultados

Las graficas obtenidas en este proyecto son los siguientes:







# Conclusiones

# Bibliografía

<https://github.com/stephenwvickers/node-net-snmp>

<http://www.fixedbyvonnie.com/2015/04/connect-your-gns3-lab-to-the-internet-mac-os-x/#.Vlyivd-rTR3>

<http://www.fixedbyvonnie.com/2015/04/connecting-a-virtualbox-machine-to-gns3-mac-yosemite/#.Vlyivd-rTR3>

<https://community.gns3.com/thread/22772>

<http://forum.gns3.net/topic3909.html>

<https://www.npmjs.com/package/net-snmp>

<http://net-snmp.sourceforge.net/tutorial/tutorial-5/commands/snmpget.html>

https://nym.se/node-snmp-native/docs/example.html

1. [↑](#footnote-ref-2)