

Virtualización mediante MetaRouter para la implementación de una red wireless de navegación anónima tipo TOR en equipos Mikrotik

Roberto David Meneses Basantes

Universidad de Fuerzas Armadas, Av. Progreso s/n, Quito-Ecuador, Teléfono: 59323989400
rdmeneses@espe.edu.ec

Darwin Leonidas Aguilar Salazar

Universidad de Fuerzas Armadas, Av. Progreso s/n, Quito-Ecuador, Teléfono: 59323989400 ext. 1869
dlaguilar@espe.edu.ec

Christian Nestor Vega Muñoz

Universidad de Fuerzas Armadas, Av. Progreso s/n, Quito-Ecuador, Teléfono: 59323989400 ext. 1872
cnvega@espe.edu.ec

Rita Paola León Pérez

Universidad de Fuerzas Armadas, Av. Progreso s/n, Quito-Ecuador, Teléfono: 59323989400 ext. 1872
rpleon1@espe.edu.ec

Resumen

Hoy en día la virtualización se ha convertido en la primera opción a la hora de considerar servidores y como tal presenta muchas ventajas sobre soluciones propietarias. Pero que tal si un administrador de red pudiera tener ambas soluciones en un dispositivo embebido el cual delegará responsabilidades, tanto a la parte virtual como a la parte física, manejando así una solución mixta.

La marca europea de equipos Mikrotik, mediante el software de sus dispositivos de capa 3 RouterOS, permite desde su versión 3.30, virtualizar mediante MetaRouter, una versión de Linux comprimido, llamado OpenWRT el mismo que proporciona un sistema de archivos totalmente modificable con gestión de paquetes, proporcionando al usuario la posibilidad de personalizar su firmware de acuerdo a su necesidad, lo que lo convierte en un sistema ideal para crear una aplicación sin que el firmware sea el limitante. OpenWRT tiene sus orígenes con la línea Wireless de cisco Lynksys la cual es una plataforma fácil de familiarizarse, con el manejo de servidores Linux basados en Debian [1].

En este documento se detallará la implementación y discutirá los resultados de un servidor de navegación anónima de tipo TOR instalado en OpenWRT y virtualizado sobre un equipo Mikrotik RB450G.

Palabra(s) Clave(s): MetaRouter, Mikrotik, OpenWRT, TOR, virtualización.

1. Introducción

The Onion Router (TOR), es un proyecto que fue lanzado el 20 de Septiembre del año 2002. Se trata de una red de comunicaciones con baja latencia, funcionando y operativa en Internet y cuya característica principal es permitir el envío de mensajes entre usuarios sin revelar su identidad (su dirección IP, anonimato desde un punto de vista de red). La Red TOR es el caso más conocido de red oscura que existe hoy en día. La Red Oscura, o darknet, se suele referir a la parte de Internet que no quiere ser procesada ni encontrada fácilmente y que suele alojar actividades ilegales [2].

La red TOR, es la mayor red de enrutamiento anónimo de tercera generación que está diseñado para realizar el intercambio de direccionamiento, escondiendo la ubicación física del usuario y tiene como uno de sus beneficios la defensa de los ataques por “análisis de tráfico”, que se realiza mediante el análisis de tramas que el host o terminal

envía y recibe desde la red.

A diferencia del enrutamiento común, que establece que se debe conocer su camino completo hasta llegar al destino antes de establecer las comunicaciones, las redes tipo TOR, utilizan para su comunicación, rutas sinuosas difíciles de seguir y que periódicamente cambian usando diferentes caminos, cada vez que se establece una nueva conexión. Los paquetes de datos de la red TOR, siguen caminos aleatorios de tres saltos encriptados, a través de varios repetidores teniendo en cuenta que cada repetidor conoce únicamente al repetidor anterior y al próximo repetidor, mas no la ruta completa. Por eficiencia TOR, utiliza la misma ruta de intercambio de datos, para las conexiones que se producen dentro de un tiempo determinado, para peticiones posteriores se traza una nueva ruta [3].

MIKROTIK es una marca proveniente de Latvia dedicada a fabricar dispositivos de networking y equipos de comunicaciones, cuenta con su sistema operativo RouterOS, que tiene funcionalidades como: manejar direccionamiento IP, Manejo de acceso al medio en capas dos y tres, redes privadas NAT, Firewall, Módulo Wireless, para motivos del presente proyecto una de las mayores virtudes de la marca, es que permite la Virtualización de Sistemas Operativos para Routers mediante códigos fuente abiertos de tipo OpenSource, entre otras ventajas, presenta las condiciones ideales para el desarrollo del presente estudio.

En el presente documento se describe y analiza los resultados de implementar una topología de red que permita navegar de forma segura sobre redes anónimas TOR. Se ha considerado, que los usuarios se encuentren detrás de una red NAT, protegiendo así su identidad, detrás de un direccionamiento privado, considerando además que el MetaRouter ejecutará el servicio TOR, enrutando así todo el tráfico web, por el MetaRouter implementado sobre un dispositivo de capa 3 con módulo Wireless, que permitirá una validación de portal cautivo de tipo Hotspot en un punto de acceso Inalámbrico (Wireless), mediante una instancia virtualizada, aprovechando el recurso de MetaRouter de Mikrotik a nivel de software. En cambio, a nivel de hardware se utilizará

placas (mainboard's) de la serie RB4XXAH con arquitectura MIPSBE, que soportan imágenes OpenWRT MetaRouter y cuentan con las características físicas necesarias para la implementación que se propone y muestra el desarrollo a continuación.

2. Desarrollo

a) Topología de la red

La topología de red considerada en la figura 1, permite la implementación de un servidor de navegación anónima tipo TOR virtualizado mediante la herramienta MetaRouter de Mikrotik, donde se obtendrá una red NAT de tráfico anónimo y tráfico tradicional. Los únicos puertos de transmisión de datos que estarán abiertos a los usuarios finales son 80 TCP, 53 UDP, 8118 TCP y 9050 TCP, el puerto 8118 TCP es el proxy Privoxy que actúa como un proxy HTTP estándar al proxy TOR Socks mientras que el puerto 9050 de TCP, es el proxy Socks disponible para el enrutamiento del tráfico a través de la red TOR.

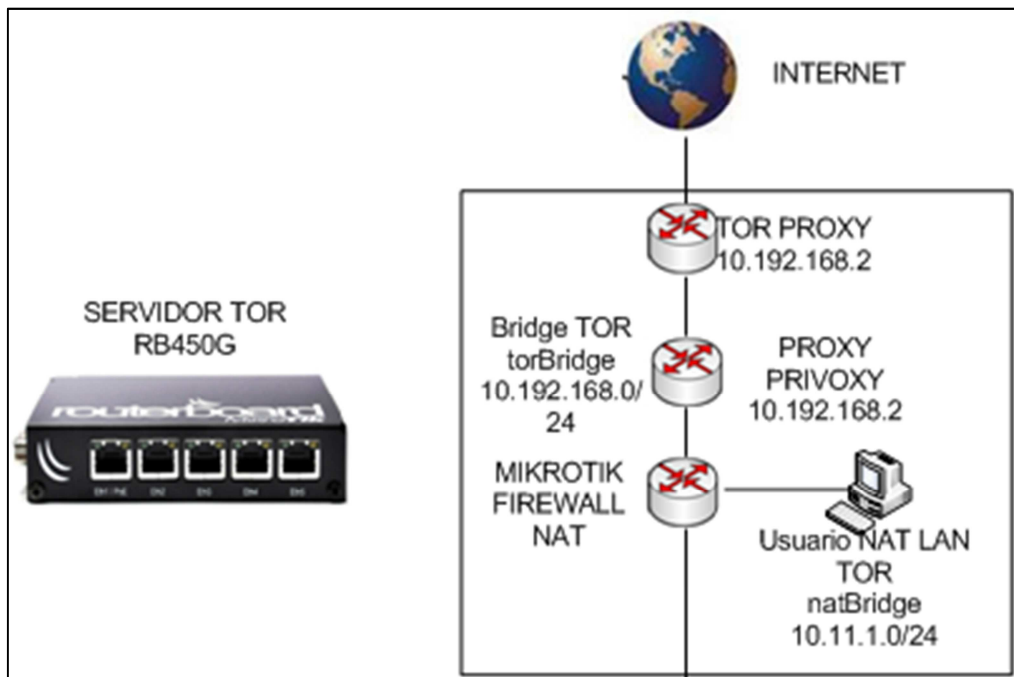


Fig. 1. Topología de la red.

b) Direccionamiento IP.

El enrutamiento considerado para las diferentes interfaces tanto del dispositivo físico como del dispositivo virtual se encuentra descrito en la Tabla 1.

Dispositivo	Interfaz	Dirección IP	Máscara de subred
RB450G	ether1	192.168.1.10	255.255.255.192
	natBridge	10.11.1.1	255.255.255.0
	torBridge	10.192.168.1	255.255.255.0
	DHCP server	10.11.1.0/24	S/N
	DNS Server	10.11.1.1	S/N
OpenWRT	Ether0	10.192.168.2	255.255.255.0
	Gateway	10.192.168.1	S/N

Tabla 1. Direccionamiento IP del servidor Mikrotik.

Es importante tener en cuenta que las interfaces de OpenWRT, son virtuales y deben ser modificadas la primera vez desde MetaRouter, como se indicará posteriormente.

c) Configuraciones.

A) Configuración sobre Mikrotik.

1. Ingreso al dispositivo

El ingreso al dispositivo se lo realizará a través del software de administración WinBox.

2. Creación de las interfaces Bridge de comunicación

La secuencia de comandos de la figura 3, configura las interfaces bridges y añaden los puertos necesarios para el bridge encargado de realizar el NAT de la red.

```
> interface bridge
/interface bridge> add name=torBridge
/interface bridge> add name=natBridge
/interface bridge> port
/interface bridge port> add interface=ether4 bridge=natBridge
/interface bridge port> add interface=ether5 bridge=natBridge
/interface bridge port> █
```

Fig. 3. Configuración de interfaces Bridge.

3. Configuración de direccionamiento IP

La configuración de direcciones IP, se deberá realizar en base a la Tabla 1 y el resultado de este deberá proporcionar la información como la mostrada en la Figura 4.

```
> ip address add interface=ether1 address=192.168.1.10/26 disabled=no
> ip address add interface=natBridge address=10.11.1.1/24 disabled=no
> ip address add interface=torBridge address=10.192.168.1/24 disabled=no
> █
```

Fig. 4. Configuración de direccionamiento IP.

4. Configuración DNS

En afán de ocultar toda la información que pueda revelar la identidad del cliente, se utilizará los servidores DNS de google, acorde a lo indicado en la figura 5.

```
> /ip dns set servers=8.8.8.8 allow-remote-requests=yes
> █
```

Fig. 5. Configuración del servidor DNS.

5. Configuración de firewall NAT Rules

Las reglas de Firewall NAT son las encargadas de redireccionar todo el tráfico que ingresa o sale a través de la interfaz WAN (ether 1) hacia el servidor virtual por su dirección IP 10.192.168.2. (Ver Figura 6)

```
> /ip firewall nat
/ip firewall nat> add chain=srcnat out-interface=ether1 action=masquerade
/ip firewall nat> add chain=dstnat in-interface=ether1 action=dst-nat to-addresses=10.192.168.2
/ip firewall nat> █
```

Fig. 6. Configuración de reglas de Firewall.

6. Configuración de la ruta por defecto

La ruta por defecto establece que todo el tráfico de red será enviado al internet a través del gateway del proveedor de internet. (Ver Figura 7)

```
> /ip route add dst-address=0.0.0.0/0 gateway=192.168.1.1
>
```

Fig. 7. Configuración de ruta por defecto.

7. Configuración del servidor DHCP sobre natBridge

El servidor DHCP, se encuentra en el pool 10.11.1.0/24, gateway 10.11.1.1 y DNS 8.8.8.8., esto se muestra en la Figura 8.

```
> /ip pool add name="nat-DHCP" ranges="10.11.1.10-10.11.1.250"
> /ip dhcp-server network add address=10.11.1.0/24 gateway=10.11.1.1 dns-server=8.8.8.8
> /ip dhcp-server add interface="natBridge" lease-time="3d" name="natDHCPserver" address-pool="nat-DHCP" authoritative=yes disabled=no
>
```

Fig. 8. Configuración del servidor DHCP.

B) Configuración sobre MetaRouter.

9. Descarga del firmware OpenWRT

Se puede descargar de manera gratuita el compilado de OpenWRT para dispositivos Mikrotik de arquitectura MIPSBE, así como todos los paquetes instalables del repositorio de Mikrotik <http://openwrt.wk.cz/>. (Ver Figura 9)

Location: /attitude_adjustment/mr-mips/

Name	Last modified	Size
Parent Directory		-
packages/		-
OpenWrt-ImageBuilder-mr-mips-for-linux-i686.tar.bz2	11-Mar-2014 13:36	-
OpenWrt-SDK-mr-mips-for-linux-i686-gcc-4.6-linaro_uClibc-0.9.33.2.tar.bz2	11-Mar-2014 13:37	317M
OpenWrt-Toolchain-mr-mips-for-mips-gcc-4.6-linaro_uClibc-0.9.33.2.tar.bz2	11-Mar-2014 13:40	591M
config-mips-aa-36088-20140311	11-Mar-2014 13:38	40M
kernel-debug.tar.bz2	11-Mar-2014 13:28	162K
kernel-debug.tar.bz2	11-Mar-2014 13:40	5.6M
mD5sums	11-Mar-2014 13:40	227
openwrt-mr-mips-rootfs-36088-basic.tar.gz	08-Mar-2014 22:28	2.2M
openwrt-mr-mips-rootfs.tar.gz	11-Mar-2014 13:28	2.3M

Fig. 9. Repositorio de Mikrotik para OpenWRT.

Una vez descargado el firmware OpenWRT Attitude Adjustment 12.09, se deberá importar el archivo .tar.gz a la carpeta files ubicada en el software WinBox. (Ver Figura 10)

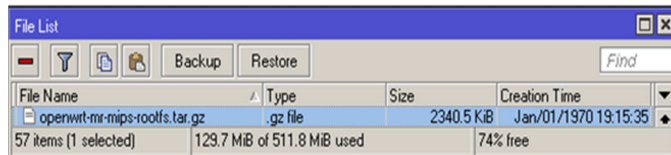


Fig. 10. Importación de la imagen de OpenWRT.

10. Creación del dispositivo virtual

El dispositivo virtual se crea a través de la herramienta MetaRouter importando la imagen descargada y asignando la memoria del dispositivo (se deberá asignar al menos 100MB de memoria para el correcto funcionamiento del servidor de navegación anónima). (Ver Figura 11)

```
admin@MikroTik] > /metarouter import-image memory-size=100 file-name=openwrt-mr-mips-rootfs.tar.gz
imported: 100%
```

Fig. 11. Creación de la máquina virtual.

11. Creación de la interfaz virtual de MetaRouter

La interfaz de OpenWRT es una interfaz de tipo dinámico y se la deberá vincular a la interfaz bridge torBridge (el bridge torBridge es un túnel de conexión directa entre el dispositivo RB450G y el dispositivo virtual), la configuración se muestra en la Figura 12.

```
> /metarouter set 0 name=attitude_tor
> metarouter interface add type=dynamic dynamic-bridge=torBridge virtual-machine=attitude_tor
```

Fig. 12. Configuración de la interfaz de OpenWRT.

12. Configuración de direccionamiento IP sobre OpenWRT

El primer paso que se deberá realizar es la configuración de la dirección IP del dispositivo virtual, de acuerdo a la Tabla 1.

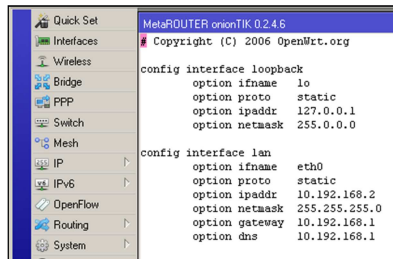


Fig. 13. Configuración de la interfaz ether0.

Para que los cambios tengan efecto se debe reiniciar el servicio de red con el comando “/etc/init.d/network restart”.

13. Habilitación de administración mediante SSH

El comando “passwd” permite habilitar el protocolo de comunicaciones SSH y automáticamente deshabilita TELNET. (Ver figura 14)

```
root@metarouter:/# passwd
Changing password for root
New password:
Bad password: too weak
Retype password:
Password for root changed by root
root@metarouter:/# passwd
Changing password for root
New password:
Bad password: too weak
Retype password:
Password for root changed by root
```

Fig. 14. Configuración de SSH.

C) Configuración sobre OpenWRT.

14. Ingreso al dispositivo virtual

El software mediante el cual se configurará y administrará OpenWRT es el demonio SSH Putty, luego se ejecutará e ingresará a OpenWRT.

15. Configuración de firewall

El compilado de OpenWRT de Mikrotik provee un pequeño Firewall el cual se deberá modificar para que acepte todo tipo de conexiones, como lo muestra la Figura 15 y después reiniciarlo “/etc/init.d/firewall restart”.

```
10.192.168.2 - PuTTY
# config defaults
option syn_flood      1
option input          ACCEPT
option output         ACCEPT
option forward        ACCEPT
# Uncomment this line to disable ipv6 rules
option disable_ipv6  1

config zone
option name           lan
option network        'lan'
option input          ACCEPT
option output         ACCEPT
option forward        ACCEPT

config zone
option name           wan
option network        'wan'
option input          ACCEPT
option output         ACCEPT
option forward        ACCEPT
option masq           1
option mtu_fix        1

config forwarding
option src            lan
option dest           wan
```

Fig. 15. Configuración del Firewall.

16. Configuración del archivo de resolución DNS

El servidor DNS deberá ser configurado desde el archivo “resolve.conf” con el siguiente comando “vi /etc/resolve.conf” (Ver Figura 16)

```
10.192.168.2 - PuTTY
# search lan
nameserver 127.0.0.1
nameserver 10.192.168.1
```

Fig. 16. Configuración del DNS de OpenWRT.

17. Configuración del módulo de actualización de librerías opkg

OpenWRT cuenta con el gestor de paquetes OPKG, que deberá ser configurado mediante “vi /etc/opkg.conf” con la dirección web del repositorio de Mikrotik visto anteriormente (Ver Figura 17).

```
10.192.168.2 - PuTTY
# rc/gz packages http://openwrt.wk.cz/attitude_adjustment/mr-mips/packages
dest root /
dest ram /tmp
lists_dir ext /var/opkg-lists
option overlay_root /overlay
```

Fig. 17. Configuración de paquete opkg.

18. Instalación/configuración módulo administración gráfico Luci

La instalación del paquete Luci se la realiza como muestra se muestra en la figura 18.

```
10.192.168.2 - PuTTY
root@Tesis:~# opkg install luci
```

Fig. 18. Instalación del paquete Luci.

Una vez instalado el paquete Luci, se deberá habilitar el paquete web uhttpd (figura 19).

```
root@Tesis:~# /etc/init.d/uhttpd enable
```

Fig. 19. Habilitación del paquete uhttpd.

El ingreso a la interfaz web, se lo realiza con el mismo usuario y clave configuradas para SSH (Figura 20).

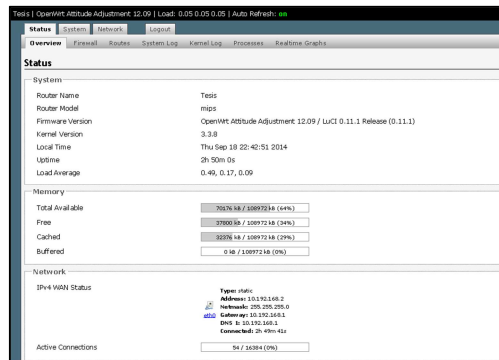


Fig. 20. Ventana del paquete Luci.

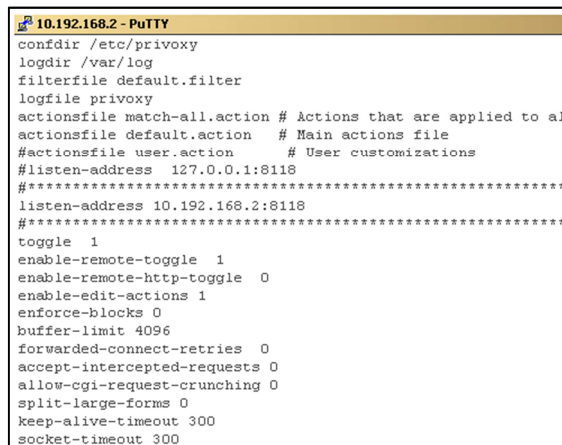
19. Instalación y configuración de Privoxy [4, 5]

El proxy Privoxy se encargará de ser el nexo entre el cliente solicitante de navegación y la red anónima TOR (Figura 21).

```
10.192.168.2 - PuTTY
root@Tesis:~# opkg install privoxy
```

Fig. 21. Instalación de Privoxy.

No existe una configuración estándar de Privoxy por lo cual esta deberá ser configurada de acuerdo a las condiciones de cada red. Privoxy deberá ser configurado para el presente escenario como lo muestra la Figura 22, en el archivo “vi /etc/privoxy/config”.



```
10.192.168.2 - PuTTY
confdir /etc/privoxy
logdir /var/log
filterfile default.filter
logfile privoxy
actionsfile match-all.action # Actions that are applied to all
actionsfile default.action # Main actions file
#actionsfile user.action # User customizations
#listen-address 127.0.0.1:8118
#*****
listen-address 10.192.168.2:8118
#*****
toggle 1
enable-remote-toggle 1
enable-remote-http-toggle 0
enable-edit-actions 1
enforce-blocks 0
buffer-limit 4096
forwarded-connect-retries 0
accept-intercepted-requests 0
allow-cgi-request-crunching 0
split-large-forms 0
keep-alive-timeout 300
socket-timeout 300
```

Fig. 22. Configuración de Privoxy.

Terminada la edición del archivo “config”, se deberá reiniciar el servicio.

20. Instalación y configuración del paquete TOR

Debido a que el repositorio de Mikrotik ofrece varias versiones del paquete de navegaciones anónimas TOR, es recomendable instalar una versión que ya ha sido probada previamente “tor-alpha_0.2.3.22-rc-1_mr-mips.ipk” (Figura 23)



```
10.192.168.2 - PuTTY
root@metarouter:~# opkg install http://openwrt.wk.cz/attitude_adjustment
/mr-mips/packages/tor-alpha_0.2.3.22-rc-1_mr-mips.ipk
```

Fig. 23. Instalación del paquete TOR.

Debido a que la memoria es el principal inconveniente en dispositivos embebidos, se creará una variable flotante que se sobrescriba cada vez que la red TOR, emita cualquier mensaje hacia el servidor (Ver Figura 24).



```
10.192.168.2 - PuTTY
root@Tesis:~# cat /dev/null >> /etc/tor/torrc
root@Tesis:~# cat /dev/null > /etc/tor/torrc
```

Fig. 24. Variable de sobre escritura torrc.

El archivo “vi /etc/tor/torrc” será configurado como lo muestra la figura 25. En este archivo está considerado que se podrá únicamente variar los parámetros de las líneas 9, 10 y 11. El resto de la configuración ha sido adecuada a la topología y

direccionamiento planteados en la Figura 1 y Tabla 1 respectivamente.

```
10.192.168.2 - PuTTY
# Archivo de configuracion y autenticacion
SocksPort 9050
SocksListenAddress 10.192.168.2:9050
SocksPolicy accept 10.11.1.0/24
Log notice file /var/log/tor/notices.log
RunAsDaemon 1
ORPort 9001
DirPort 9030
ExitPolicy reject *:
Nickname TesisTor
RelayBandwidthRate 100 KB
RelayBandwidthBurst 200 KB
DNSPort 53
DNSListenAddress 10.192.168.2
AutomapHostsOnResolve 1
```

Fig. 25. Configuración de TOR.

Para que se produzca el intercambio de información con la red TOR el archivo TOR debe ser reiniciado, como se indica en la figura 26.

```
root@Tesis:~# /etc/init.d/tor restart
```

Fig. 26. Reinicio del servicio de TOR.

Al finalizar la configuración se podrá visualizar un mensaje de escucha de los puertos con los que opera la red TOR (ver figura 27).

```
Oct 21 02:06:29.455 [notice] Opening Socks listener on 10.192.168.2:9050
Oct 21 02:06:29.459 [notice] Opening DNS listener on 10.192.168.2:53
Oct 21 02:06:29.464 [notice] Opening OR listener on 0.0.0.0:9001
Oct 21 02:06:29.467 [notice] Opening Directory listener on 0.0.0.0:9030
root@metarouter:~#
```

Fig. 27. Notificación de puertos de TOR.

21. Configuración de navegación mixta en el cliente

Al ser una solución de navegación mixta, se deberá configurar el proxy en el navegador que se utilizará para acceder de forma anónima, como lo muestra la Figura 28.

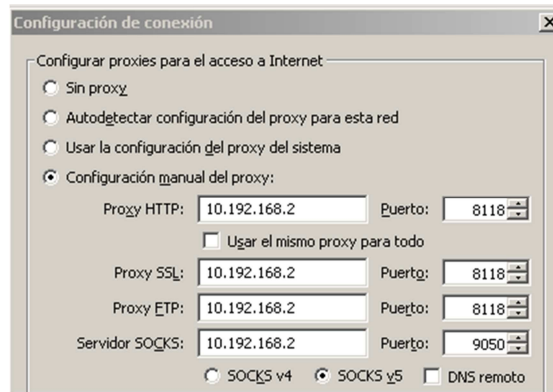


Fig. 28. Configuración del proxy en el navegador.

22. Navegación anónima

Una vez finalizada la configuración, se podrá verificar si está o no navegando en la red TOR mediante la página web www.check.torproject.org. (Ver Figura 29)

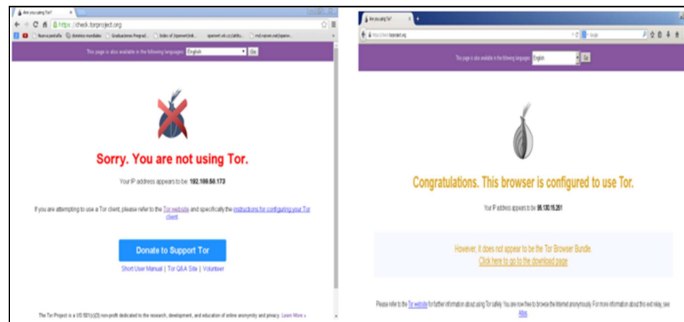


Fig. 29. Página de verificación de navegación.

Se podrán observar, dos resultados dependiendo del navegador utilizado, si el navegador es en el que se configuró previamente el proxy, se podrá navegar en la red TOR, si se utiliza otro navegador se podrá hacer únicamente uso de internet de la manera tradicional.

3. Resultados

A) Medición de ancho de banda.

Para poder realizar una comparación de Ancho de Banda, se utilizó un enlace de datos de 3 Mbps con una compartición 8:1 (ADSL dato del proveedor CNT - ECU). (Figura 30)

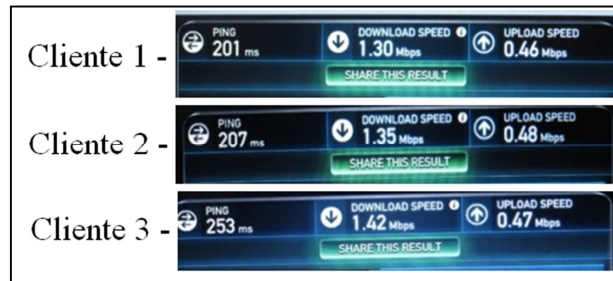


Fig. 30. Medición de ancho de banda para 3 clientes.

En la Figura 30, se realizó la medición de ancho de Banda con tres clientes simultáneos a través del mismo nodo de salida en la red TOR.

Las gráficas de resultados fueron tamizadas a partir de tres escenarios de evaluación, descritos en la Tabla 2.

Escenario	Dispositivos	
	LAN	Wireless
1	1	1 Tor – 1 Tradicional
2	2	1 Tor – 1 Tradicional
3	3	1 Tor – 1 Tradicional

Tabla 2. Escenarios de evaluación del servidor.

B) Comportamiento del dispositivo físico Mikrotik.

Se puede verificar que tanto las características físicas del servidor como las características de los servicios implementados se encuentran dentro de un rango estable de operación, esto puede visualizarse en las figuras 31 y 32 respectivamente.

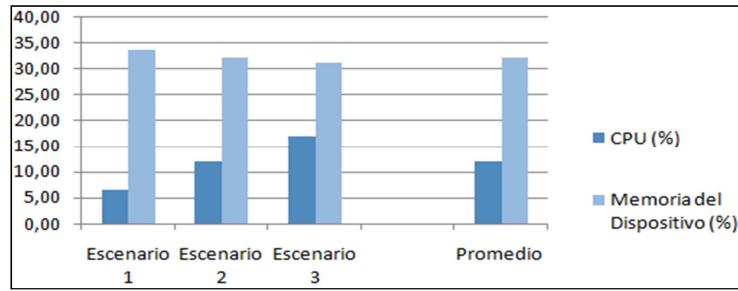


Fig. 31. Uso del CPU y memoria de Mikrotik.

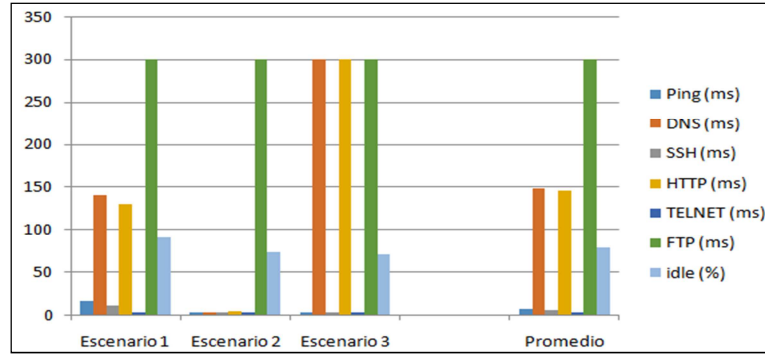


Fig. 32. Tiempos de conexión a servicios de Mikrotik.

C) Comportamiento del dispositivo virtual OpenWRT.

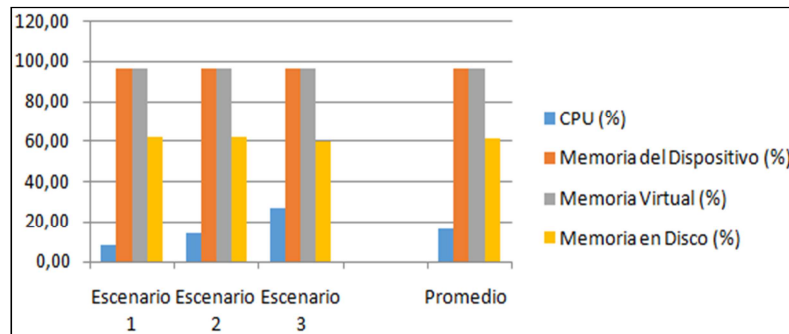


Fig. 33. Uso del CPU y memoria de OpenWRT.

El consumo de recursos generado por el firmware virtual, no representa una carga de procesos superior a la carga del dispositivo físico, permitiendo que tanto el dispositivo físico como el virtual puedan coexistir en un mismo entorno de red, esto es plenamente concluyente a partir de la figura 33.

D) Medición de ancho de banda.

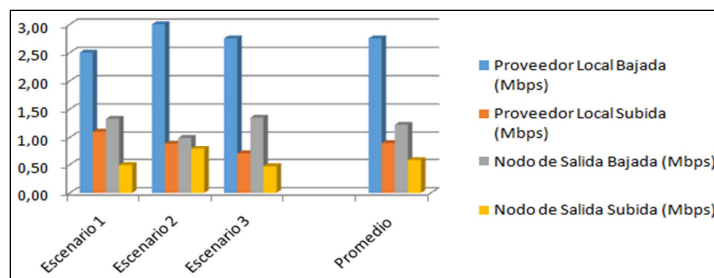


Fig. 38. Medición de ancho de banda en TOR.

De acuerdo a la prueba de ancho de banda se verifica que la navegación anónima, provee menor velocidad de navegación, sin embargo el tener una solución de navegación mixta permitirá aprovechar la velocidad de la navegación tradicional y el anonimato de la red TOR, compensando de alguna manera la pérdida de velocidad en la red TOR.

4. Discusión

Mediante las pruebas realizadas a la topología de red implementada, se evidencia que el servidor de navegación virtualizado mediante MetaRouter, provee navegación anónima sobre el pseudo-dominio .onion de la red TOR, sin comprometer el funcionamiento del servidor Mikrotik, sobre el cual se realizó la virtualización ni tampoco la interacción con los demás dispositivos de comunicación que operan sobre la red.

El aplicar este tipo de herramientas (configuraciones) puede potencialmente permitir que los usuarios evadan ciertos sistemas de control o autenticación ya que su aplicación involucra el desconocer la identidad de los usuarios, potencialmente podrían ser utilizadas para aplicaciones de accesos no autorizados a redes de terceros.

Es muy importante, el grado de seguridad que se quiere implementar, sobre una red de comunicaciones, especialmente en afán de prevenir, problemas tan frecuentes como ladrones de identidad, suplantadores, vendedores de información y demás estafas a las

que se está expuesto en internet. Uno de los problemas, más básicos de privacidad, se produce cuando, un atacante ajeno a la transmisión de datos, se infiltra en la comunicación mediante medios físicos, teniendo acceso, a la información y con ello a las cabeceras y datos embebidos en ellos. Pero también existen formas mucho más potentes, para infiltrarse en la comunicación desde fuera de la red local y realizar un análisis de tráfico, muchos atacantes espían múltiples puntos de internet, rastreando patrones de comunicaciones comunes entre emisores y receptores, en esos casos la protección usando encriptación, no es suficiente en la protección de la información, ya que únicamente esconde los datos del tráfico de internet, mas no las cabeceras. Los campos donde se puede aplicar estos tipos de bloqueo son sumamente amplios, ya que van desde la protección económica, al prevenir ataques bancarios y uso de información indebida, hasta la protección personal, al no revelar la ubicación geográfica de cada usuario.

No menos importante sería el tener en cuenta que con fines de realizar delitos informáticos, sería muy conveniente y útil el usar estas técnicas de “disfraz”. Sin embargo, la investigación actual de la ciencia debe tender o proponer la generación de códigos y herramientas que puedan manejar técnicas de predicción “aprendidas” de este tipo de software y hardware a fin de que estas potenciales aplicaciones no sean utilizadas con fines delictivos.

5. Conclusiones

Los resultados de las pruebas muestran que la navegación anónima sobre la red TOR es aproximadamente dos tercios más lenta que la navegación tradicional, esto se debe al algoritmo de funcionamiento de TOR el cual es independiente del entorno de red sobre el cual esta implementado el servidor de navegación anónima.

Debido a que la red TOR maneja únicamente tráfico TCP, la conversión mediante socks de los demás protocolos a TCP aumenta el tiempo de petición de comunicación a la red

TOR, lo que a su vez se ve reflejado en el aumento de los tiempos de respuesta de la navegación web.

El nivel de exigencia al que está sometido el servidor RB450G, cuando la red ya está en producción no sobrepasa el 65% del uso del procesador, lo que lo convierte en una opción válida y de bajo costo a la hora de implementar servicios virtualizados en pequeñas y medianas redes de comunicaciones.

El firmware OpenWRT permite añadir características y herramientas mediante paquetes de instalación que firmwares propietarios no proveen, permitiendo potenciar un equipo, lo que a su vez brinda la posibilidad de implementar un firmware personalizado a medida de las necesidades de la red.

Independientemente del uso comercial que pueda representar esta solución de firmware mixta, el aprovechar los recursos de un firmware virtual sobre un firmware propietario en un dispositivo de bajo costo, representa una nueva opción de aprendizaje y entrenamiento en sistemas de OpenSource en entornos académicos, aunque se deberá tener especial atención con el número de usuarios que accederán a través del proxy Privoxy ya que el procesamiento del dispositivo Mikrotik se incrementa y esto involucraría la afectación a la velocidad de acceso a la red y la disminución del ancho de banda ofertada.

6. Referencias

- [1] MikroTik RouterOS. MikroTik. Letonia. 2010.
- [2] J. I. Cerón Bergantiños, "Onion routing y Red Tor". Universidad Politécnica de Madrid Escuela Técnica Superior de Sistemas Informáticos. 2014.
- [3] Tor:Overview. Disponible: <https://www.torproject.org/about/overview.html>. Julio 2015.

[4] Tor Stable Manual. Disponible: <https://www.torproject.org/docs/tor-manual.html>. Julio 2015.

[5] RouterBOARD 450G Series User's Manual. MikroTik. Letonia. 2009.

7. Autores

Ing. Roberto Meneses Basantes, Ingeniero Electrónico en Redes y Comunicación de Datos, graduado el año 2014 en la Universidad de las Fuerzas Armadas en Quito – Ecuador. Sus trabajos principalmente están relacionados con seguridad basada en aplicaciones Open Source.

Ing. Darwin Aguilar Salazar, Ingeniero Electrónico en Telecomunicaciones y Master en Redes de Comunicaciones. Docente principal de la Universidad de las Fuerzas Armadas. Coordinador del área de Redes de Información. Sus estudios de investigación están alineados con las vulnerabilidades y análisis de rendimiento eficiencia en redes de datos.

Ing. Christian Vega Muñoz, Ingeniero Electrónico en Telecomunicaciones. Docente principal de la Universidad de las Fuerzas Armadas. Jefe del Laboratorio de Networking, su investigación está relacionada en la optimización de la comunicación de datos sobre las Redes de Transporte.

Ing. Paola León Pérez, Ingeniera Electrónica en Automatización y Control y Master en Diseño, Producción y Automatización Industrial. Docente principal de la Universidad de las Fuerzas Armadas. Sus estudios e investigaciones están alineadas con dispositivos FPGA y Microelectrónica.