ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES

"IMPLEMENTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE USUARIOS CON API Y CONTROL DE ANCHO DE BANDA BASADO EN POLÍTICAS QUEUE TREE EN UN HOTSPOT INALÁMBRICO MIKROTIK"

Trabajo de titulación

Tipo: Propuesta tecnológica

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

AUTOR:

DARYEL ISRAEL LEON CACHOTT

Riobamba-Ecuador

2021

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES

"IMPLEMENTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE USUARIOS CON API Y CONTROL DE ANCHO DE BANDA BASADO EN POLÍTICAS QUEUE TREE EN UN HOTSPOT INALÁMBRICO MIKROTIK"

Trabajo de titulación

Tipo: Propuesta tecnológica

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

AUTOR: DARYEL ISRAEL LEON CACHOTT

DIRECTOR: ING. MARCO VINICIO RAMOS VALENCIA MGs.

Riobamba-Ecuador

2021

Yo, Daryel Israel Leon Cachott, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación. El patrimonio intelectual pertenece a l Escuela Superior Politécnica de Chimborazo

Riobamba, 11 de julio de 2021

Daryel Israel Leon Cachott

0604180661

ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA DE INGENIERA EN ELECTRÓNICA, TELECOMUNICACIÓNES Y REDES

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación: Tipo: Propuesta tecnológica, **"IMPLEMENTACIÓN DE UN SISTEMA DE ADMINISTRACIÓN DE USUARIOS CON API Y CONTROL DE ANCHO DE BANDA BASADO EN POLÍTICAS QUEUE TREE EN UN HOTSPOT INALÁMBRICO MIKROTIK"**, de responsabilidad del señor: **DARYEL ISRAEL LEON CACHOTT**, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

	FIRMA	**FECHA**
Ing. Jefferson Ribadeneira Ramirez **PRESIDENTE DEL TRIBUNAL**	Firmado electrónicamente por: JEFFERSON ALEXANDER RIBADENEIRA RAMIREZ	.....
Ing. Marco Vinicio Ramos Valencia **DIRECTOR DE TRABAJO DE TITULACIÓN**	Firmado electrónicamente por: MARCO VINICIO RAMOS VALENCIA	.....
Ing. Alberto Leopoldo Arellano Aucancela. **MIEMBRO DEL TRIBUNAL**	ALBERTO LEOPOLDO ARELLANO AUCANCELA Firmado digitalmente por ALBERTO LEOPOLDO ARELLANO AUCANCELA Fecha: 2021.10.18 19:09:41 -05'00'	.....

DEDICATORIA

Mi motivación y la principal razón de cada día querer salir adelante han sido mis padres, a los cuales dedico el presente trabajo como culminación de una etapa y un logro más obtenido, por haber sido mi fortaleza en los momentos más duros de vida y por apoyarme en cada peldaño de mis estudios. Por ellos he puesto mi esfuerzo en este trabajo para devolverles un poco de todo lo que me han dado. A mi hermana Andrea porque ha sido mi ejemplo de superación personal y a mi hermana Alanys porque cada día intento ser el mejor ejemplo para ella. A Dios por darme la sabiduría y bendecirme con una familia y un hogar lleno de amor. A todos ellos dedico el presente trabajo.

AGRADECIMIENTO

Agradezco a Dios por regalarme un día más de vida y por haberme otorgado la capacidad intelectual necesaria para llegar a este punto. A mis padres por siempre confiar en mí, apoyarme en mis estudios y por siempre estar exigiendo un poquito más de mí.

El agradecimiento especial es para todos y cada uno de mis docentes que han impartido sus conocimientos y sin los cuales este logro no hubiera sido posible, por haber tenido la capacidad y paciencia para sembrar el conocimiento en mí.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE ANEXOS

RESUMEN

En el presente trabajo se realizó la aplicación de la API (Interfaz de Programación de Aplicaciones) de Mikrotik para realizar un manejo eficiente en la creación de credenciales de usuario en un Hotspot. Se estudió las políticas de calidad de servicio que mejor se acoplaron a las necesidades de los clientes en la red, para posteriormente realizar la comparación de los paramentos de evaluación obtenidos, como ancho de banda, latencia, jitter y pruebas PING; obteniendo que, al aplicar un control de ancho de banda con políticas de calidad (QOS), la red obtiene una evidente mejoría. Posteriormente con respecto a la eficiencia en la creación de credenciales de usuario, se obtuvo que, a diferencia de la alternativa manual, con la interfaz gráfica mediante la API el tiempo de creación de credenciales es menor, obteniendo así una mejor eficiencia en el tiempo que toma dicha acción. Además, evidenciando que a medida que el número de credenciales requeridas aumenta, el tiempo que toma en desarrollar la creación también, comparando con la alternativa de la API. Después de la implementación a través del routerboard rb951ui-2hnd se pudo comprobar la aplicación de las políticas de calidad de servicio en usuarios reales, por último, se evaluó a través del perfil pre configurado, que se respeten los tiempos de acceso a internet para la posterior eliminación automática del usuario con tiempo de conexión agotado. Comprobando que cada usuario puede acceder a la red el tiempo por el cual está solicitando el recurso con un ancho de banda establecido y asegurando una calidad de servicio a través de la priorización de tráfico. Se recomienda establecer un ancho de banda acorde al número de usuarios que llegará a albergar la red ya que de ser muy escaso repercutirá en la calidad de la experiencia que tendrán los usuarios.

Palabras clave: <INTERFAZ DE PROGRAMACIÓN DE APLICACIONES (API)> <HOTSPOT> <ROUTERBOARD> <ANCHO DE BANDA> <CALIDAD DE SERVICIO>.

ABSTRACT

The API (Application Programming Interface) of the Mikrotik application was carried out in the current work to perform efficient management in the creation of user credentials in a Hotspot. The quality of service policies that best adapted to the needs of the clients in the network was studied to make a comparison of the evaluation parameters obtained, such as bandwidth, latency, jitter, and PING tests; obtaining that, when bandwidth control is applied with quality policies (QoS), the network gains an evident improvement. Subsequently, regarding the efficiency in the creation of user credentials, the result was that, unlike the manual alternative, with the graphical interface through the API the credential creation time is less, thus obtaining a better efficiency in the time than take such action. In addition, it was possible to show that as the number of required credentials increases, the time it takes to develop the creation as well, compared with the API alternative. After the implementation, by using the rb951ui-2hnd RouterBOARD it was possible to verify the application of the quality of service policies in real users. Finally, the internet access times are respected for the subsequent automatic deletion of a user with a connection timed out was evaluated through the pre-configured profile. Checking that each user can access the network for the time for which the resource is requesting with an established bandwidth and ensuring a quality of service through the prioritization of traffic. It is recommended to establish a bandwidth according to the number of users that the network will host since if it is very scarce, it will affect the quality of the experience that users will have.

Keywords: <APPLICATION PROGRAMMING INTERFACE (API)> <HOTSPOT> <ROUTERBOARD> <BANDWIDTH> <SERVICE QUALITY>.

INTRODUCCIÓN

Para llegar a cubrir la necesidad de las personas por el recurso del internet existen diferentes alternativas, una de ellas es el alquiler del recurso por un tiempo limitado a cambio de un valor fijado, es ahí donde, siguiendo la tendencia del avance de la tecnología, a través de un Hotspot se puede realizar esta acción de manera inalámbrica donde el usuario pueda acceder el tiempo que desee a la conexión de internet desde cualquier dispositivo móvil.

La administración de una red de cualquier tipo muchas veces llega a ser un proceso para personas capacitadas, ya que se requiere de un conocimiento ya sea en los equipos en los que se va a trabajar o bien en su software, sin embargo, con el pasar de los años esta tarea se ha ido simplificando mediante softwares para que cualquier persona sin una previa capacitación pueda manejar su red, a través de interfaces gráficas más amigables y mucho más intuitivas que merman el proceso complejo. El administrador no necesita saber cómo funciona el proceso sino más bien verlo aplicado lo más pronto posible.

A pesar que el avance de la tecnología brinda la posibilidad de conexiones inalámbricas, esto también puede llegar presentar ciertos desperfectos, por lo que para asegurar al usuario una calidad de servicio, es indispensable la incorporación de políticas de calidad de servicio, además de garantizar un servicio por el cual se está pagando. Es ahí donde el control de ancho de banda es necesario para la distribución equitativa del recurso.

ANTECEDENTES

En la actualidad el Internet se puede considerar un servicio básico en todos los hogares, representa una herramienta fundamental para el desarrollo académico, por lo que, con el fin de cubrir la demanda del mismo y satisfacer las necesidades de personas sin los recursos necesarios, existen sistemas inalámbricos de internet que el municipio de la ciudad de Riobamba pone a disposición, sin embargo, los mismos están solamente en ciertos puntos y la calidad de servicio ofrecido se puede cuestionar. Para dicho objetivo existe en el mercado un sin número de marcas disponibles en equipos enrutadores, entre las cuales Mikrotik representa una opción muy asequible con más funciones, está ampliamente disponible en el mercado y de fácil obtención. Mikrotik cuenta con un sistema operativo propietario RouterOS, el mismo que es altamente personalizable además de flexibilidad con respecto a la posibilidad de realizar diferentes enfoques. (Pedron, 2018)

Otro tipo de sistemas son los denominados ‘cyber café’, que ponen a disposición de la población computadoras de escritorio con internet las cuales pueden ser utilizadas durante el tiempo que el

usuario lo necesite a cambio de un valor tarifado según el tiempo de uso. A pesar que pueden llegar a ser una muy buena opción, no están disponibles en todos los sectores que lo necesitan, y además del hecho de no tener privacidad y estar en un entorno junto con otras personas, haciendo uso de equipos compartidos, podría representar una vulnerabilidad a la privacidad del usuario. Otro punto cuestionable seria la calidad de servicio que se está recibiendo, ya que el propietario o dueño no pone a disposición de sus usuarios este tipo de información y de esta forma los mismos no pueden exigir un servicio por el cual es pagando. Por último, las opciones de servicios fijos de internet en casa o planes para dispositivos móviles suelen escalar a valores de precios muy elevados que pueden llegar a no ser asequibles para todas las personas, ya que sus sistemas de basan en mensualidades que el usuario final debe cancelar sin falta, a cambio de contar con un servicio fijo, además de requerir instalaciones con costos elevados para implantación del servicio en el hogar. Debido a la gran demanda actual de la población por el servicio de internet es que se considera necesaria la implementación de un sistema capaz de ofrecer un servicio de calidad, el mismo que podría ser utilizado en un dispositivo personal, sin la necesidad de instalaciones ni ataduras de cables. Un sistema inalámbrico que provea internet que satisfaga con las necesidades que el usuario final requiere.

FORMULACÍON DEL PROBLEMA

¿Se puede implementar un sistema de administración de usuarios mediante API y aplicar políticas de Queue Tree para el manejo de ancho de banda en un Hotspot?

SISTEMATIZACIÓN DEL PROBLEMA

¿Es posible aplicar políticas de QoS a un sistema de Hotspot?
¿Sería posible implementar un sistema automatizado de generación de credenciales de usuario para acceso a la red inalámbrica?
¿Se puede obtener un mejor rendimiento en un sistema Hotspot a través de políticas de QoS y automatización de generación de credenciales de acceso?

JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN

JUSTIFICACIÓN TEÓRICA

La implementación de este tipo de sistemas de alquiler de servicio de internet a cambio de una tarifa fija por un periodo determinado de tiempo se remonta a los años 90, donde comenzó a tomar fuerza la popularidad de los denominados ‘cyber café’, y se extendió a lo largo de la década de

los 2000. Sus inicios fueron en Inglaterra donde comenzaron como lugares de conexión permanente a internet a cambio de una pequeña tarifa donde se podría intercambiar mensajes con amigos y familiares. Pero, no es hasta la masificación del internet, durante la primera década del siglo XXI que comienzan a tornarse muy populares y representar un buen negocio. En Ecuador este tipo de sistemas tiene su nicho bien establecido, sin embargo, en la actualidad con los nuevos sistemas de internet fijo en el hogar y dispositivos Smartphone la popularidad de los sistemas de 'cyber café' poco a poco se vuelven obsoletos y no se ha optado por un avance tecnológico para mejorar el servicio en base a las necesidades de los usuarios. Por lo que una actualización a este tipo de sistemas, adaptándolo a las necesidades actuales de la población por un servicio de internet de calidad, de fácil y rápido acceso sin ataduras de cables, por periodos determinados de tiempos, podría representar una manera eficaz de cubrir con la demanda del servicio de internet en lugares sin cobertura. En la ciudad de Riobamba actualmente no se registra implementaciones de este tipo de sistemas, y en ciertos sectores sin cobertura podría ser de mucha ayuda para cubrir con la demanda del servicio de internet

JUSTIFICACIÓN APLICATIVA

Este proyecto surge de la necesidad de encontrar nuevas alternativas a las convencionales de alquiler de servicio de internet, conservando todas las características que las mismas ya poseen e incluyendo nuevas funcionalidades que conlleven a una mejor experiencia al usuario. Dichas funcionalidades incluyen principalmente la aplicación de árboles de colas para asegurar una calidad de servicio al usuario. Además de facilitar la administración de la red a la persona a cargo, automatizando procesos durante la administración de usuarios.

Entre los dispositivos que se proponen utilizar están: Mikrotik routerboard rb951ui-2hnd con sistema operativo Router OS utilizado para la creación del Hotspot, manejo de usuarios de la red, implementación de QoS. Para la creación automática de usuarios y claves dependiendo de los perfiles que se configure en el router se va a utilizar la API Mikrotik PHP mediante servidor XAMPP. El portal cautivo en los dispositivos conectados a la red será el provisto por el router en la creación y configuración del Hotspot

OBJETIVOS

OBJETIVO GENERAL

- Implementar un sistema de administración de usuarios con API y controlar el ancho de banda basado en políticas Queue Tree en un Hotspot inalámbrico Mikrotik.

OBJETIVOS ESPECÍFICOS

- Estudiar las políticas de QoS aplicadas a un sistema Hotspot.
- Diseñar e implementar la interfaz gráfica de manejo automatizado de credenciales de acceso a la red para los usuarios mediante la API de Mikrotik.
- Evaluar el rendimiento de la aplicación de políticas Queue tree sobre la disponibilidad de ancho de banda para los usuarios.
- Analizar la eficiencia de la administración de credenciales implementada para el acceso a la red de Hotspot.

MÉTODOS Y TÉCNICAS

MÉTODOS

- Método exploratorio: Recopila, organiza y analiza la información acerca del problema o tema planteado que se pretende estudiar para seguir con una investigación más exhaustiva.
- Método experimental: Obtener los datos mediante pruebas con diferentes administradores de usuarios de Hotspot para examinar los resultados obtenidos con el fin de identificar y comprender de mejor manera en funcionamiento del sistema.
- Método Inductivo: Teniendo en cuenta los datos alcanzados se logrará valorar el desempeño de la red y generar conclusiones generales partiendo de los antecedentes que tratan acerca de los sistemas de redes inalámbricas para acceso a internet.

TÉCNICAS

- Revisión documental: Conseguir información de varias fuentes acerca de sistemas inalámbricos Hotspot para acceso a internet.
- Sistematización: Permite emplear y ordenar la información recopilada de forma crítica y reflexiva para el análisis de resultados de los datos obtenidos.
- Análisis y síntesis: Tiene como fin evaluar y mostrar los datos significativos y a continuación se realizarán las conclusiones pertinentes acerca del sistema.

CAPÍTULO I

1. MARCO TEÓRICO

1.1 Introducción

En la actualidad las redes inalámbricas juegan un rol fundamental en el día a día de las personas, ya sea, para acceder a una red móvil o para acceder a internet desde un dispositivo inalámbrico, esta tecnología abolió el uso de cables remplazando por medios no guiados. Las redes inalámbricas están presentes en todo lugar y cotidianamente estamos expuestos a ondas electromagnéticas, algo que en un principio no resultaba así, ya que, tecnologías no cableadas eran destinadas más para el ámbito empresarial, pero el pasar de los años y el avance de la tecnología, nos ha acercado más a un mundo más inalámbrico.

Con la incursión cada vez más fuerte de redes inalámbricas, una gran problemática que se presento fue vulnerabilidad en la seguridad, es por tanto que existen dos escenarios plenamente definidos, el primero un escenario de hogares, donde la seguridad no representa un punto fundamental más allá de tener protegida con una contraseña para evitar la intromisión de personas ajenas a la red, mientras que, para un ambiente empresarial, la seguridad juega un papel importante, además de manejar siempre el proceso más eficiente que se logre, es por tanto que en el ámbito empresarial se puede llegar a optar por servidores de seguridad, y herramientas que automaticen procesos a través de interfaces más amigables con el usuario, una solución válida representan las API's, con las cuales a través de un lenguaje más conocido se puede llegar a automatizar procesos actuando como un intermediario virtual.

1.2 API (Application Programming Interface)

Una API se puede entender como el intermediario entre dos servicios de software, facilitando que las funcionalidades del uno puedan ser utilizado por el otro para mejorar los resultados. Sirve de enlace entre un software creado y otro al cual este le puede ser útil, es decir una interacción Software to Software. (Plaza, S. E, Ramírez, N. L, Acosta, C. M, 2016).

La creación e implementación de una API, debe exponer claramente las funcionalidades que va a ofrecer, dejando en claro cuál será el uso para quien desee hacerlo. Entre los beneficios observables está la posibilidad de obtener cada vez los mismos resultados en diferentes escenarios sin perder tiempo en re implementaciones. Es así que se llega a obtener una automatización

1.2.1 Elementos de una API

Los dos elementos principales que son imprescindibles en todas las Apis de servicio web son: los datos y los métodos. A pesar que dependiendo del tipo API pueda existir una variación en cuanto a las especificaciones de estos elementos, en todas sin excepción se implicarán datos y métodos, con la diferencia que en ciertos tipos de API´s, se requerirán claves o autenticaciones o en tal caso ambas.

- **Datos API.**

El formato de los datos que se proporciona a través de la mayoría de API´s suele ser XML, JSON, RSS, Atom y PHP serializados. Las API´s que ofrecen más de una opción de formato facilitan una mayor flexibilidad en cuanto al desarrollo de la misma.

- **Métodos API**

La manera en como los desarrolladores manipulan los datos a través de una API principalmente suele ser mediante solicitudes HTTP GET y solicitudes POST.

1.2.2 Tipos de API´s

Existe diferentes tipos de API´s, como primera clasificación está según a quien están distribuidas. Las mismas que son: (IBM, 2016)

- Internas o privadas: En este caso los únicos beneficiaros de la utilidad son los mismo que generaron el código, es decir será utilizada la API solo dentro de la misma compañía o asociación. El principal objetivo es mejor la productividad en sus propios intereses sin llegar a beneficiar a los demás.

- Externas o públicas: Se puede llegar a obtener las mismas ventajas o beneficios que una privada, al no tener que volver a programar desde cero el código, pero en este caso está al servicio de todo aquel que desee hacer uso de la misma.

- Partner: Este tipo representa una colaboración entre varias asociaciones o compañías, por lo tanto, suponen un punto medio entre privadas y públicas. De tal manera que el código

generado por una empresa, podría ser útil para otra y viceversa, con esto ambas llegan a beneficiarse, pero sin aportar directamente a la productividad de la competencia.

Sin embargo, no es la única clasificación, analizando otro parámetro, en este caso a que están orientadas, son: (Maddox, 2016).

- API´s de servicios Web: Como el nombre lo indica hace uso de una dirección web para proporcionar el acceso al servicio requerido.

- API´s basadas en librerías: Su gran importancia se basa en la practicidad al momento de programar en determinado lenguaje, ya que se puede incorporar mediante librerías, claro está que no es la única manera ya que el programador tiene la posibilidad de generar sus propias librerías basadas en sus necesidades.

- API´s de sistemas operativos: Pensadas principalmente para poder determinar cómo están estructuradas las funcionalidades de un sistema operativo.

1.2.3 API´s basadas en librerías

Las API´s son importantes para las librerías porque les permiten conectarse con el ecosistema web más amplio e integrarse con otros servicios web destacados de requerirlo. Actualmente las API´s permiten que las librerías se unan en esa interconexión. Además de este beneficio tienen la capacidad de cambiar los flujos de trabajo de la librería y mejor los servicios existentes.

Para API´s basadas en librerías, dependiendo el código fuente con el que se vaya a llevar a cabo existiría la librería indicada, algunos ejemplos de lenguajes de programación que cuentan con librerías son:

- PHP
- Delphi
- C/++/#
- Python
- Ruby on rails
- VB.Net
- Java
- Perl

1.2.4 API Mikrotik

La API de Mikrotik ofrece la funcionalidad de llevar a cabo aplicaciones mediante la comunicación con el sistema operativo de Mikrotik, Router OS. Logrando así administrar equipos que dispongan de este sistema operativo.

El puerto utilizado para la comunicación, API – Router, es el TCP 8728, cabe señalar que por defecto el servicio de API esta deshabilitado en las configuraciones del Router.

La aplicación ejecutándose a través de un servidor web, permite la conexión remota mediante las credenciales para acceder a las funcionalidades, en la Figura. 1-1. Se puede observar la arquitectura para la conexión con la API de Mikrotik.

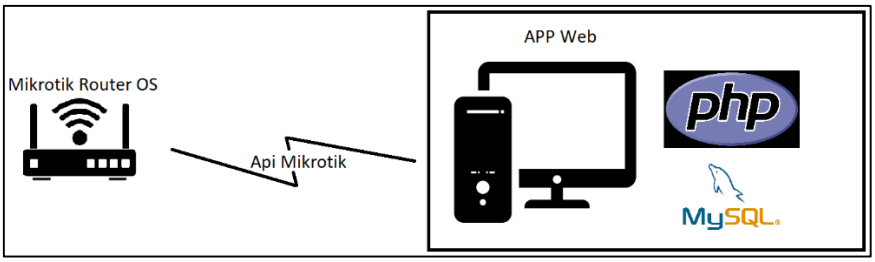


Figura 1-1: Entorno virtual de conexión con Api Mikrotik

Realizado por: Leon Daryel, 2021

Mikrotik a través de su página oficial pone a disposición la versión 1.6 de la librería implementada para código PHP. Versión desarrollada por los autores Nick Barnes y Denis Basta. (Barnes, 2015)

A través del lenguaje de programación PHP se logra el envío y recepción de comandos hacia dispositivos que cuenten con el sistema operativo Router OS. Adema PHP ofrece la posibilidad de adaptar nuevos requerimientos de manera dinámica. (PHP Group, 2017). Por lo que se optó por el desarrollo sobre PHP.

1.2.3.1 Principales características de PHP

- Lenguaje orientado a objetos
- Independencia de plataforma
- Sintaxis similar a códigos anteriores (C, C++)

- Código abierto
- Velocidad
- Robustez

Un documento PHP consta básicamente de dos partes importantes: Código HTML, donde se especifican estilos y diseños que hagan falta para la interfaz gráfica, pueden ser propiamente atributos HTML u hojas de estilo CSS.

Por otra parte, están las instrucciones PHP que son las encargadas de cumplir con: Recepción y manipulación de datos de formularios que pueden ser interiores o exteriores, toma de decisiones en función de algún evento o dato, comunicación de datos hacia implicaciones HTML, entre otras acciones que puede desempeñar instrucciones PHP. Para instrucciones PHP se debe seguir la sintaxis que se puede observar en la Figura. 2-1. Donde se muestra las señalizaciones necesarias al momento de creación de código PHP.

```
<?php // inicio de código php
instrucciones; // Todas acabadas en ';'
?> // final de código php
```

Figura 2-1: Sintaxis de código de programación PHP

Fuente: Arias Miguel, 2013

Todas las instrucciones precedidas de los símbolos ¨//¨ son tomadas como comentarios en el código de programación. Al igual que en todos los lenguajes de programación, PHP hace uso de variables a su manera, así como la le recepción de valores o establecimiento de información a una constante, la cual posteriormente puede ser interpretada por código HTML.

```
---------------------------------------------------------------

•  <h3>
•
•  <?php
•  $nombre = $_POST['nombre']; // Recoge el valor
•  $texto = "Hola, "; // Establece un texto fijo
•  echo $nombre. $texto; // Contenido interpretable por
(X)HTML
•  ?>
•
•  </h3>

---------------------------------------------------------------
```

Figura 3-1: Sintaxis para el manejo de variables en PHP

Fuente: Arias Miguel, 2013.

Una variable en PHP puede ser denominada de manera libre, empezando siempre por el símbolo ¨$¨ para ser seguido de una letra o nombre de la variable. En código PHP no es necesaria agregar una distinción a la variable, con el simple hecho de asignar un valor a la variable esto determina del tipo que será.

Una desventaja de PHP es el hecho de no poderse ejecutar en el ordenador del usuario, para que PHP puede ejecutarse es necesario tener instalado un servidor de PHP

1.3 Seguridad en redes inalámbricas de área local

Las redes inalámbricas son aquellas que carecen de un medio tangible de comunicación es decir sin cables, el método de comunicación es mediante medios no guiados a través de ondas electromagnéticas, haciendo uso para esto de antenas. En una red inalámbrica un emisor puede poseer una antena y llegar a varios receptores, pudiendo emplear más de una en el caso de una comunicación en ambos sentidos. Cuando la distancia entre el emisor y receptor es demasiado extensa la señal podría verse afectada por la disminución de potencia o atenuación, por lo que se hace uso de antenas intermedias denominadas repetidoras para llegar a cubrir mayores distancias las cuales regeneran la señal enviada originalmente.

Una red inalámbrica con respecto a una convencional presenta una instalación más rápida, a menores costos y con una movilidad mayor, lo que conllevaría a una mayor productividad, sin embargo, también posee ciertas desventajas como por ejemplo una mayor tasa de errores, al no tener un medio físico la velocidad y las interferencias aumentan y que la seguridad podría verse comprometida.

Al igual que en redes convencionales con cable, dentro de las redes inalámbricas se puede identificar distintos tipos según el alcance de la cobertura

Las amenazas informáticas día a día van evolucionando y en aumento, en la actualidad hay una gran diversidad de amenazas como puede ser: malware, virus, troyanos, phishing, etc. Y a pesar que no existe únicos métodos para combatir las amenazas y que ningún sistema es complemente seguro hay técnicas comunes para poder evitar ciertos tipos de amenazas. Como medidas

preventivas una seguridad de la información conlleva realizar análisis de posibles riegos en puntos fundamentales.

- **Autenticación**

Cuando un cliente solicita acceso a un servidor un punto crucial es la exigencia de la autenticación del usuario, A nivel de capa de enlace de datos los protocolos utilizados para la autenticación están:

- PAP (Pasword Authentication Protocol): Utilizado para autenticar a un cliente hacia un servidor, forma parte como sub protocolo de PPP. No se puede considerar un protocolo muy seguro al transmitir la contraseña sin cifrarse.
- CHAP (Challenge Handshake Authentication Protocol): Como su nombre lo indica CHAP utiliza desafíos, lo que significa exigir al cliente que desea acceder que demuestre su identidad. La verificación de realiza en base a un secreto compartido que sería la contraseña, a diferencia de PAP trasmite solo una parte de la contraseña por lo que se considera un método más seguro.

- **Cifrado**

Una vez superada la autenticación se prosigue a una serie de métodos para codificar la información, existen dos métodos según el tipo de clave los cuales se podría utilizar:

- Cifrado de clave simétrica: Comúnmente utilizados en usuarios individuales o sistemas cerrados ya que el método de descifrado debe ser provisto por el emisor hacia el receptor por anticipado. Esto podría aumentar el riesgo ya que para la trasmisión mediante inclusión de terceros se puede originar alteraciones, sin embargo, como ventaja representa la agilidad del método asimétrico.
- Cifrado de clave asimétrica: A diferencia de una cifrado simétrico se manejan dos tipos de claves una publica y una privada las cuales servirán en el descifrado de la información. Al ser dos tipos de claves, la publica puede ser compartida de cualquier manera mientras que la clave privada es la que se deberá mantener en secreto.

1.3.1 Control de acceso

El control de acceso es un método que permite garantizar que los usuarios prueben ser quien dicen ser, pudiendo ser a través de la autenticación de credenciales. Basándose en donde se introduce el control de acceso a la red, existen tres estrategias.

- **Control en el perímetro**

El control se lo realiza en el exterior, es decir e un punto donde los usuarios se conectan a la red, un ejemplo sería el switch en una red LAN.

- **Control central**

En esta estrategia el control se puede implementar en cualquier punto de acceso a la red, por ejemplo, en un punto medio de la red donde se coloque un dispositivo por el que pasa el tráfico que requiera acceso.

- **Control en el cliente**

Todo el control o políticas de seguridad que sea necearías se lo realiza en el usuario final, es decir instalando en los equipos del usuario final en el cual se requiera la gestión y control de aplicaciones.

Como método de control de acceso a los usuarios antes de que puedan acceder a los servicios proporcionados en una red local (LAN), existe el estandar IEEE 802.1x.

1.3.2 Protocolo 802.1x.

Es un protocolo que proporciona control de acceso y autenticación y se basa en la arquitectura cliente servidor, creado por IEEE e inicialmente fue destinado para redes alámbricas, pero con el tiempo se extendió este protocolo hacia redes inalámbricas. Dentro del protocolo 802.1x tres participantes juegan un papel importante:

- El suplicante: Es el dispositivo que desea conectarse a la red inalámbrica
- El servidor: Contiene información necesaria para identificar que dispositivos están autorizados para tener acceso a la red. El protocolo 802.1x fue diseñado para hacer uso de servidores RADIUS

- El autenticador: Es el dispositivo por el cual el cliente hace la petición para conseguir acceso a la red, este podría ser un switch o enrutador al cual tenga conexión el dispositivo suplicante.

El proceso de la autenticación se lleva a cabo mediante el protocolo EAP a través del servidor RADIUS. En el caso de un acceso inalámbrico el servidor RADIUS hace uso de claves WEP dinámicas usadas durante el proceso de cifrar la conexión entre el cliente y el servidor, el mismo servidor es el que se encargará de cambiar dichas claves WEP de manera dinámica con el fin de evitar ataques de acceso a la red por fuerza bruta. El protocolo EAP que emplea certificados de seguridad presenta variantes:

- EAP-TLS: El método de autenticación es mutuo de manera que el servidor autentica al cliente y viceversa, por lo que requiere instalación de certificados tanto en el cliente como en el servidor. El cifrado es a través de protocolo TLS y soporta el uso de claves dinámicas WEP.
- EAP-TTLS: A diferencia de EAP-TLS solo se requiere instalación de certificados en el servidor no en el cliente, proporcionando servicios similares, sin embargo. La instalación de certificados en el servidor garantiza una autenticación fuerte del servidor desde el cliente y por otro lado la autenticación de cliente por parte de servidor se llevará a cabo una vez establecida la sesión TLS, a través de un protocolo como PAP, CHAP o sus variantes.
- PEAP: De igual manera solo requiere certificado en el servidor, la protección se da a través del establecimiento de un túnel seguro TLS entre cliente y el servidor.

1.4 Hotspot

Es un tipo de red inalámbrica con puede constar de uno o varios puntos de acceso, con el fin de obtener acceso a internet generalmente. Un Hotspot puede cubrir zonas donde no existe tendido de cable para internet o cubrir sitios públicos de afluencia de gente donde se puede ofrecer el servicio de manera gratuita o previo pago. El manejo de usuarios dentro de un Hotspot es a través de un protocolo de autenticación donde los usuarios con credenciales podrán obtener acceso. Mientras que por parte del cliente generalmente se usa una interfaz donde el usuario deberá ingresar las credenciales. A este tipo de interfaces de las conoce como Portal Cautivo, donde además se puede llegar a especificar términos de uso y condiciones.

1.4.1 Portal cautivo

Es una página web a través de la cual los usuarios que deseen tener acceso a una red inalámbrica deben obligadamente ingresar antes de poder obtener recursos de dicha red. El usuario que este ingresando no puede evitar este tipo de interfaces antes de obtener acceso.
Al ser una página web por lo general se hace uso de lenguaje PHP o HTML la misma que deberá estar localizada en un servidor o localizada en un punto de acceso trabajando conjuntamente con el protocolo de autenticación de usuarios.

1.4.2 Administración de usuarios

Un portal cautivo maneja un proceso con el cual cumple el objetivo de la administración de usuarios, siguiendo una secuencia obligatoria de pasos implementada mediante el siguiente desgrama.

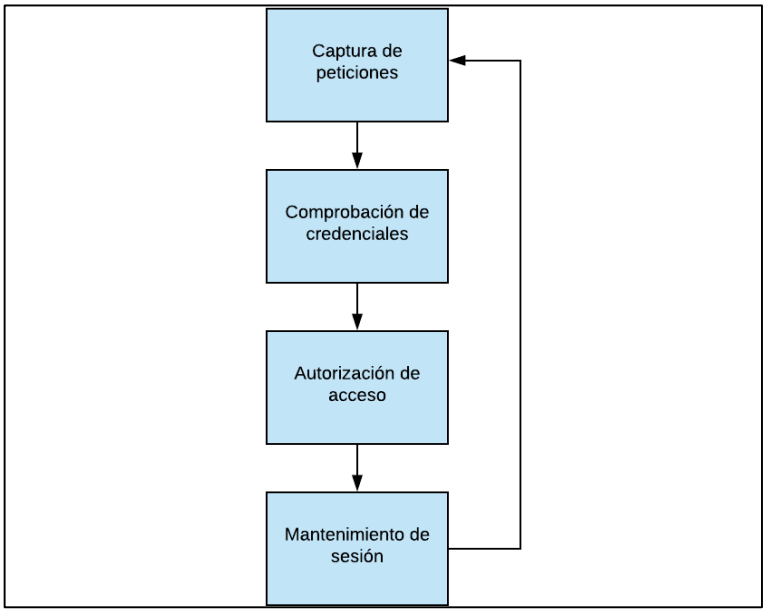


Figura 4-1: Proceso de control de usuarios en portal cautivo

Realizado por: Leon Daryel, 2021

El uso de claves para los clientes incrementa el nivel de seguridad para el acceso, de esta forma si un usuario logra acceder al dispositivo inalámbrico (AP), pero no posee credenciales de acceso, no podrá hacer uso de ningún recurso es decir ni recibir ni enviar tráfico.

1.5 Calidad de servicio sobre internet (QoS).

En redes informáticas, el término utilizado se lo conoce como QoS (Quality of Service). Mismo que hace alusión al desempeño de la red el cual es apreciado por los usuarios miembros de dicha red.

Para medir la calidad del servicio por lo general se consideran algunos aspectos o indicadores que muestran el desempeño, tales como, latencia, jitter, perdida de paquetes, anchos de banda, etc. Estos aspectos son particularmente importantes durante la transmisión de tráfico con requerimientos especiales, tal como podría ser servicios de Streaming.

A través de la implementación de una calidad de servicio, es posible proveer diferentes prioridades a diferentes tipos de tráfico de que se podrían generar en la red. De esta manera garantizando un mínimo nivel de rendimiento a un flujo de tráfico. Dentro de redes con capacidades limitadas o insuficientes, el garantizar la calidad de servicio es importante, en aplicaciones de transmisión multimedia, ya que este tipo de tráfico puede ser muy sensible al *retraso.*

1.5.1 Indicadores de desempeño en una red

En el transcurso desde el origen hasta el destino, se pueden presentar diferentes problemas en los paquetes, los puntos detallados a continuación son indicadores de dichos problemas que se podrían llegar a presentar.

- **Retardo.**

Este indicador refleja el periodo que el paquete toma en alcanzar su destino, puede ocurrir que se tome un largo periodo de tiempo, debido a problemáticas que se presenten como, existencia de largas colas de espera, toma de rutas más largas hacia el destino, congestión de la red. Sobre ciertos tipos de tráfico como por ejemplo voz sobre IP o Streaming, dichos retardos podrían estropear la aplicación haciéndola inutilizable.

- **Latencia.**

De manera similar la latencia es producida cuando existe demora en el envío de paquetes dentro de una red, con la diferencia que el indicador latencia, representa la suma de todos los retardos en tiempo que se producen dentro de la red.

- **Jitter.**

El Jitter en una red es un indicador de la variación en el tiempo de retardo entre paquetes, representa que existe fluctuación en el retardo a medida que los paquetes a traviesan la red. La problemática que representa esta en la interrupción de la secuencia del envío normal de paquetes.

- **Ancho de banda.**

La garantía de la disponibilidad de ancho de banda suficiente es especialmente necesario en servicios principales transmitidos a través de un enlace, aun cuando este llegue a estar ocupado. Realizando un control del recurso, se puede mejorar la utilización del ancho de banda y evitando un agotamiento.

- **Perdida de paquetes.**

Cuando un paquete enviado desde el origen no logra culminar su camino a través de la red se produce la pérdida del mismo. Dependiendo de la aplicación que envía los paquetes se puede considerar una tasa aceptable de perdida de paquetes, de esta manera perder un numero bajo de paquetes puede no causar daño en el mensaje final, sin embargo, a medida que el porcentaje de perdida de paquetes aumente, el mensaje podría llegar en cierto punto a ser incomprensible.

1.5.2 Enfoques principales de QoS en redes IP

Algunos modelos o mecanismos propuestos por la IETF (Internet Engineering Task Force). En su RFC. 3644, incluyen los que se denomina, IntServ, DiffServ, Best-Efford. Cada uno con sus características en base a los indicadores antes estudiados como retardos, latencia, perdida de paquetes, ancho de banda, etc.

- **IntServ**

Su modelo se basa en un enfoque parametrizado donde, se establece una reservación de recursos, cuando se realiza la solicitud de recursos a lo largo de la red, para la reserva del recurso utiliza el protocolo RSVP (Reservation Protocol). Como problemáticas de aplicación está la escalabilidad.

- **DiffServ**

Modelo basado en la división de tráfico mediante diferentes clases y prioridades, por lo que a diferencia de IntServ, implementa el modelo priorizado. No existe reservación de recursos,

además de lograr reducir la carga en la red. Es más escalable y como desventaja que presenta esta la no garantía en los servicios al no existir reservas de recursos. (Dobladez. 2007).

- **Best-Efford**

Este tipo de mecanismo representa un intento de la red de hacer que el tráfico generado en el transmisor, logre llegar al destino, sin garantizar la certeza de éxito del dicho envío.

1.6 Mecanismos de calidad de servicio (QoS)

La calidad de servicio dentro de redes de datos desempeña la función de controlar el tráfico para poder lograr un mejor rendimiento con respecto al canal de comunicación, logrando así disminuir tiempos de latencia y un mejor uso del ancho de banda. El control se lo lleva a cabo a través de la priorización o retrasado de paquetes. Para esto se utiliza lo que se denomina colas o ¨Queue¨ en inglés las mismas que pueden ser de tipo simple o un árbol de colas. Logrando almacenar el tráfico hasta que se pueda procesar de manera serializada.

El proceso de clasificación aplicando calidad de servicio puede usar muchos criterios: clasificación por equipo de destino, por paquetes, aplicación, etc. Sin un proceso de clasificación el concepto de calidad de servicio no puede ser considerado. La clasificación busca a que parámetro de QoS previamente negociados pertenece cada paquete del tráfico generado. Este proceso se puede observar en la Figura. 5-1.

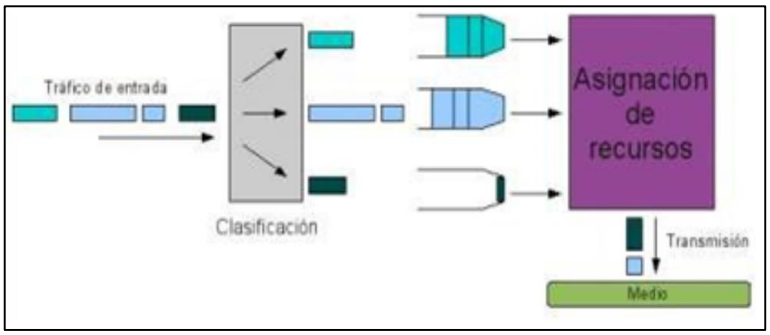


Figura 5-1: Esquema para el análisis de QoS

Fuente: Cruz Felipe. 2013.

Las políticas de calidad de servicio aplicadas al Hotspot se basan en control de ancho de banda para garantizar un mínimo del recurso y priorización de tráfico.

1.7 Control de ancho de banda basado en árbol de colas (Queue Tree).

En una red Hotspot lo principal es el control del ancho de banda hacia los usuarios conectados, ya que, sin el mismo, un usuario podría abarcar todo el recurso provocando aumento de retardos en los demás usuarios, RouterOS de Mikrotik realiza la implementación de políticas de calidad de servicio (QoS) a través del algoritmo HTB el cual basa su estructura en colas jerárquicas con diferentes prioridades, de esta manera creando relaciones entre colas padre y colas hijo.

1.7.1 Algoritmo HTB (Hierarchical Token Bucket).

La estructura creada por el algoritmo HTB de colas padres y colas hijas. Asegura una jerarquía de manera que una cola solo se convertirá en padre cuando posea al menos una hija, las mismas que consumirán el tráfico y se encargan de satisfacer la propiedad "Limit-at" que es el ancho de banda mínimo que debe cubrir. Mientras que las colas padres se encargan de repartir el recurso y una vez asegurado el ancho de banda mínimo reparten el resto.

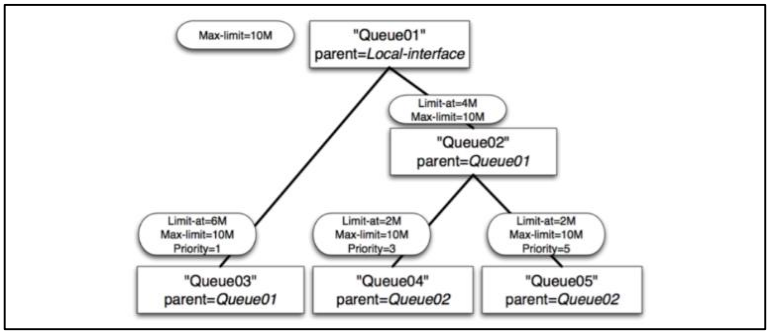


Figura 6-1: Estructura de algoritmo HTB

Fuente: Diaz Juan, 2008.

Como se puede observar en la Figura. 6-1. A partir de una cola padre se pueden dividir varias colas hijo las misma que serán padre de otras colas hijo en el caso de tenerlas. La propiedad "Max-limit", será el ancho de banda máximo que una cola es capaz de proveer y cuya cantidad se reparte entre las colas hijas, de tal forma que la suma del ancho de banda de todas las hijas de una cola padre debe ser menor o igual al ancho de banda máximo.

En una cola hijo, por su parte maneja dos tipos de campos, el "Limit-at" y de igual manera el "Max-limit". Campos necesarios en los cuales se especifica el ancho de banda mínimo que otorgara, y el ancho de banda máximo que se podría repartir en el caso de haber el recurso respectivamente. Bajo condiciones normales el campo "Limit-at" es el encargado de suministrar esa cantidad de ancho de banda, mientras que, bajo condiciones especiales, es decir, cuando la

cola padre tiene disponible un mayor ancho de banda del especificado en el “Limit-at” de la cola hijo comenzara a aplicar el “Max-limit” donde de esta forma reparte todo el ancho de banda disponible en ese momento de esta manera sin desperdiciar el recurso.

Para la implementación de calidad de servicio en un dispositivo Mikrotik es decir mediante HTB, se debe seguir una serie de pasos necesarios.

- Creación de colas padres de subida (Upstream) y bajada (Downstream)
- Creación de colas hijas de subida (Upstream) y bajada (Downstream)
- Marcación de paquetes
- Marcación de conexiones

1.7.2 Colas simples (Simple Queue)

Proporcionan un control sencillo del tráfico de la red, y las reglas se basan de prioridad por IP. No se puede asignar anchos de banda específicos para ICMP por lo que, si en un momento el ancho de banda del usuario cliente está completo, los tiempos de ping aumentaran, así como los tiempos de espera.

Por la sencillez no es recomendable utilizar en redes conde exista varias reglas o donde ese comparta el canal ya que al ser cada cola simple el método de cumplimiento será una por una según el orden configurado. Está pensado para conexiones punto a punto para priorizar direcciones IP

1.7.3 Árbol de colas (Queue Tree)

Un árbol de colas representa una estructura más compleja que una simple cola, permite una clasificación por paquetes, por protocolo, por puerto y además por el tipo de conexión. Se crean estructuras de colas de manera jerárquica con relación entre ellas a manera de padres e hijos.

A diferencia de una cola simple y además la ventaja que ofrece un árbol de colas es que se puede asignar anchos de banda ICMP, consiguiendo así que, a pesar de un usuario tener al máximo el consumo del ancho de banda asignado, el tiempo de ping aún se mantenga estable. Sim embargo por lo general un árbol de colas es un método más complicado de configurar con respecto a un sistema de colas simples.

1.7.4 Formas de encolamiento

La manera predeterminada de una cola por lo general es FIFO, por sus siglas en ingles ¨First in First out¨ lo que hace referencia a que el tráfico que entra primero será el tráfico que primero logrea salir. Esta forma de cola no requiere configuración extra. Lo que hace es simplemente procesar y reenviar los paquetes que se recibe en el orden en que llegan. Un problema que se presenta es que si la cola llegara a saturarse descartara los paquetes nuevos que se intenten enviar. FIFO representa una manera insuficiente para aplicaciones en tiempo real bajo condiciones de congestión, ya que no existe preferencias ni discriminación de paquetes, por lo que existen formas de colas de mayor eficiencia para tráfico de alta prioridad:

1.7.4.1 PFIFO, BFIFO, MQ PFIFO

RouterOS soporta varios tipos o formas de encolamientos, las tres formas se basan en el algoritmo FIFO con sus pequeñas diferencias. PFIFO utiliza un sistema de medición basado en número de paquetes, mientras que BFIFO se basa en medición por bytes. Al utilizar todos, el algoritmo FIFO, cada paquete que no pueda llegar a ser encolado o en el caso que la cola este llena, se va a descartar.

Otra característica de estos tres tipos es la mejor utilización del canal, aunque llega a presentar incrementos en la latencia. Por otra parte, MQ PFIFO es una variante de PFIFO con la única diferencia de un soporte de múltiples colas de transmisión

Como se puede observar la Figura. 7-1. La diferencia entre las formas de encolamiento se basa en la medición utilizada, cuando el tamaño de la cola es menor al límite definido en el tipo de la cola este se encolará, y por otra parte cuando el tamaño de la cola supere el limite definido, se va a descartar.

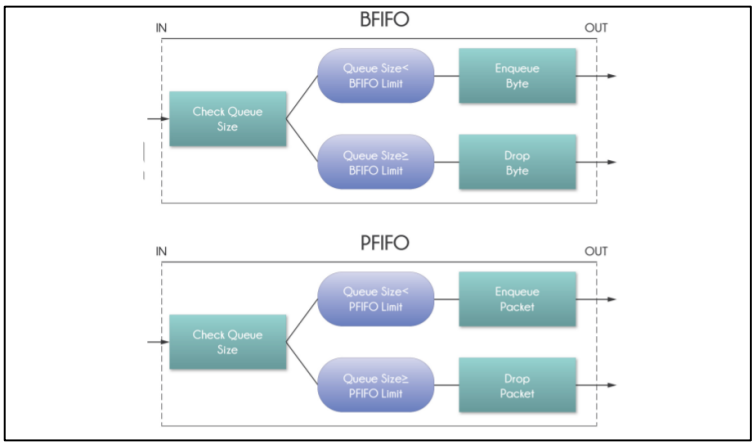


Figura 7-1: Análisis encolamientos basados en FIFO

Fuente: Escalante Mauro, 2016

1.7.4.2 Red

Este tipo de encolamiento viene del inglés ¨Random Early Drop¨, cuyo mecanismo se basa en evitar congestiones a través del control promedio de la cola. (Figura. 8-1). Por lo que para dicho control se establecen umbrales, un mínimo y un máximo, la siguiente figura muestra probabilidades donde un paquete podría ser descartado.

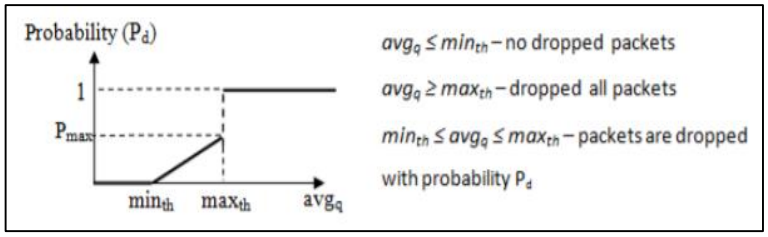


Figura 8-1: Diagrama de probabilidad de descarte de paquetes en RED

Fuente: Escalante Mauro, 2016

Si el tamaño promedio de la cola (avgq), está por debajo del umbral mínimo, ningún paquete será descartado, mientras que, para un tamaño promedio de la cola mayor al umbral máximo, todos los paquetes entrantes serán descartados. Por otro lado, si el promedio de la cola se encuentra entre el umbral mínimo y máximo, el descarte de los paquetes entrantes se lo realizara al azar con probabilidad Pd. Donde la probabilidad es exacta en función del tamaño de la cola, de la siguiente manera:

$$Pd = Pmax(avgq - minth)/(maxth - minth)$$

Donde Pmax representa el radio que puede ajustar la probabilidad de rechazar los paquetes abruptamente, comúnmente este campo toma el valor de 1 por simplicidad.

La forma de encolamiento en RED como se mencionó se basa en el tamaño promedio de la cola, a partir del mismo en base al umbral mínimo y máximo establecido se toma la decisión de encolar o descartar. Sin embargo, en un punto medio jugará un papel importante la probabilidad, a continuación, se muestra en la Figura. 9-1 el procesamiento que se realiza en un paquete de ingreso.

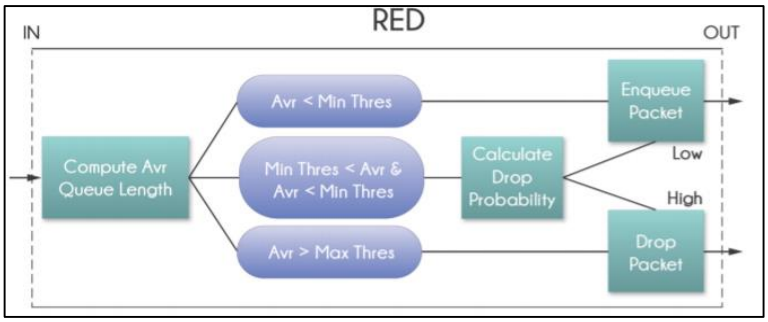


Figura 9-1: Modo de operación RED

Fuente: Escalante Mauro, 2016

1.7.4.3 SFQ

Las siglas SFQ provienen del inglés (Stochastic Fairness Queuing), esta forma de encolamiento utiliza el algoritmo de Hash para identificar un flujo de tráfico a través de 4 parámetros: Dirección de origen (src-address), dirección de destino (dst-address), puerto de origen (src-port) y puerto de destino (dst-port). La manera de clasificar los paquetes es colocándolos en uno de los 1024 sub-streams posibles. Además de el algoritmo de Hashing, también utiliza el algoritmo Round-Robin para hacer la distribución del ancho de banda disponible en todos los 1024 sub-streams. Como se puede observar en la Figura. 10-1. Dado un flujo entrante lo primero que se realiza es la clasificación, para proceder a la distribución del ancho de banda una vez realizada la clasificación. Una cola completa puede contener hasta 128 paquetes, teniendo en cuenta que hay 1024 sub-streams disponibles

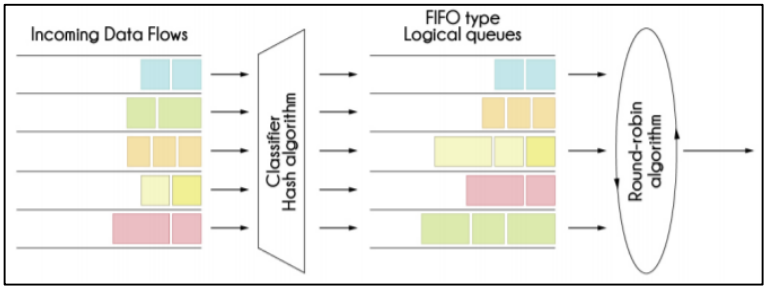


Figura 10-1: Modo de encolamiento SFQ

Fuente: Escalante Mauro, 2016

El tipo SFQ a diferencia de los previos tipos de encolamiento, no utiliza el método de asignar una cola a un flujo de datos, sino el algoritmo que maneja es capaz de dividir el trafico entrante de datos en un número determinado de colas, enumeradas de 0 (0x000) a 1023 (0x3FF).

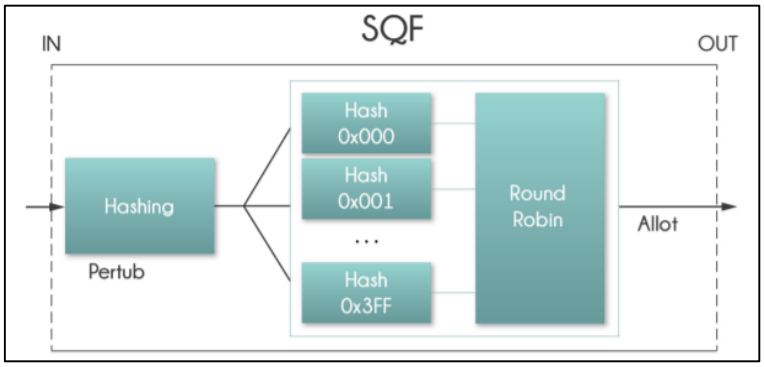


Figura 11-1: División de tráfico a través de algoritmo Hashing

Fuente: Escalante Mauro, 2016.

1.7.4.4 PCQ

Este tipo de encolamiento funciona de manera similar a SFQ, pero con funciones adicionales, proviene del inglés (Per Connection Queuing). Entre las funciones adicionales está el hecho de poder elegir identificadores de flujo, de manera que cada sub-stream será asignado a un cliente en particular. Además de la opción de limitar la velocidad a los sub-stream a través del campo pcq-rate donde si el campo toma un valor de 0 se dividirá de manera equitativa el trafico entrante.

PCQ fue diseñado para ofrecer una calidad de servicio en servicios masivos, donde se manejan colas iguales para los distintos sub-streams. El algoritmo PCQ, utiliza clasificadores para distinguir los diferentes sub-streams, a partir de aquí se aplica un tamaño de cola FIFO con su respectiva limitación en cada uno de los sub-streams, para por ultimo englobar en una cola y aplicando a la misma una nueva limitación y tamaño. Gráficamente expresado en la Figura. 12-1

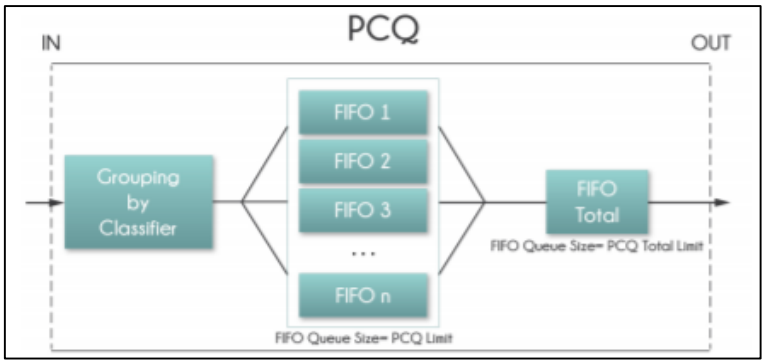


Figura 12-1: Modo de encolamiento PCQ

Fuente: Escalante Mauro, 2016.

En PCQ se manejan parámetros importantes de configuración entre los que están:

- Burst-rate: Máxima tasa de datos permitida de subida o bajada que puede llegar a ser alcanzada
- Burst-threshold: Valor de burst en el switch
- Burst-time: Periodo de tiempo sobre el cual se calcula la tasa de datos promedio.
- Classifier: Identificador para los sub-stream
- Limit: Tamaño de cola para cada sub-stream
- Rate: Máxima cantidad de datos disponibles en cada uno de los sub-stream
- Total limit: Tamaño total de la cola global de todos los sub-streams

La técnica de englobar todos los sub-streams en uno solo global permite que, en lugar de tener una gran cantidad de colas con cierta limitación, se pueda tener una sola cola en PCQ donde albergue una cantidad de sub-streams.

1.7.5 Marcación de paquetes y conexiones

En un enrutador Mikrotik el "Mangle" resulta un tipo de marcador para el procesamiento de paquetes, por ejemplos al establecer colas o "Queues", una marca identifica al paquete dentro del enrutador ya que dichas marcas no si transmiten a lo largo de la red.

El enrutador consta de interfaces de entrada y de salida, y en dicho proceso existe 5 lugares o lo que se denomina parámetro "Chain" donde se puede colocar marcas.

- Prerouting: Las reglas o marcas en este tipo de "Chain" se aplican a los paquetes al llegar a la interface de entrada.
- Input: Las reglas o marcas en este tipo de "Chain" se aplican a los paquetes justo antes de ingresar al proceso local.
- Output: En este tipo de "Chain" las reglas o marcas se aplican a los paquetes justo después que han sido producidos por un proceso.
- Forward: Las reglas o marcas en este tipo de "Chain" se aplican a un paquete que ha sido enrutado a través del dispositivo actual.
- Postrouting: En este tipo de "Chain" las reglas o marcas se aplican a paquetes cuando abandonan la interfaz de salida.

Para un mayor entendimiento del lugar de marcado la Figura. 13-1 muestra gráficamente en donde ocurriría el marcado según el tipo de “Chain” que se seleccionaría.

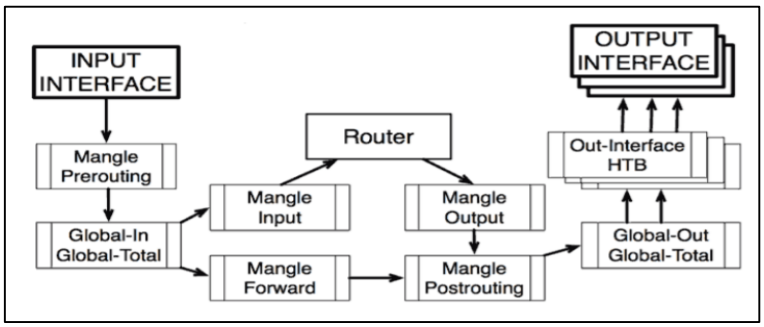


Figura 13-1: Lugar de marcación de paquetes

Fuente: Clep. Mario, 2009.

Las marcas que existen, además, se pueden colocar a un paquete o a una conexión, la diferencia primordial se basa en el hecho que, en un marcado de conexión lo que se etiqueta es la misma, es decir el camino por donde atravesaría el tráfico generado, de esta manera se realiza el marcado una sola vez in tener que especificar obligatoriamente la fuente o destino, mientras que, para un marcado de paquetes, es indispensable marcar las dos direcciones es decir de ida y venida.

1.8 Priorización de tráfico basado en árbol de colas (Queue Tree)

Al tener marcación de tráfico se puede dar prioridad a uno u otro tipo de tráfico según su necesidad, de esta manera los protocolos de más prioridad se los especifican dentro de una escala del 1 al 8 donde 1 será la máxima prioridad, en la Figura. 14-1. Se puede observar un diagrama de los protocolos con sus respectivas prioridades, donde dependiendo de las necesidades de la red se puede especificar uno u otro número de prioridad.

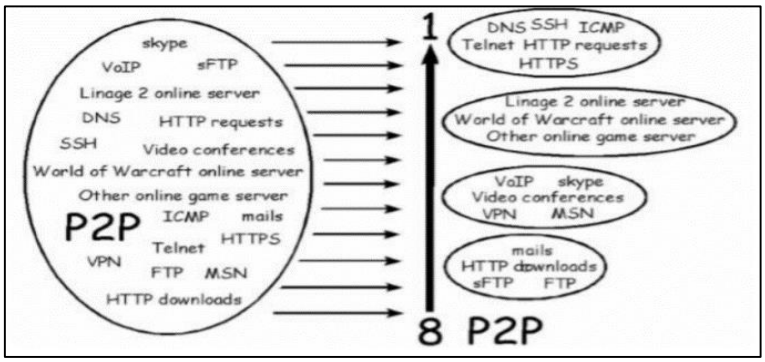


Figura 14-1: Priorización de tráfico según el protocolo

Fuente: Diaz Juan, 2008.

- Protocolo ICMP: Principalmente utilizado para comprobar si un Host está disponible o no, en cualquier petición ya sea UDP o TCP se utiliza al recibir dicha petición en un puerto que no está a la escucha, ICMP no realiza un consumo elevado de ancho de banda, pero lo que se debe asegurar son tiempos bajos de respuesta, por lo que siguiendo el esquema de la Figura. 16-1. Se debe priorizar con el máximo valor a este protocolo.
- Protocolo UDP: Proporciona un servicio orientado a datagramas, sin una garantía que los mismos lleguen a su destino y tampoco que el orden de llegada sea el orden en el que se envió, es un protocolo más simple y menos fiable, pero por lo mismo más veloz, y sus aplicaciones se basan en datos que requieran ser transmitidos los más rápido posible. Por este hecho, al trafico UDP se lo debería priorizar por debajo del protocolo ICMP.
- Protocolo TCP: A diferencia de UDP es más confiable y se orienta a la conexión, por lo que se asegurar un arribo de paquetes sin error, y de conservar el orden con el que salieron de la fuente. Por estas características se puede disminuir la prioridad del tráfico generado bajo este protocolo, trafico WEB, sin embargo, cabe señalar que el trafico WEB seria el que abarque más consumo de ancho de banda, por lo que la prioridad disminuiría, pero la asignación de dicho recurso debería incrementar.

La red Hotspot está orientada a una conexión temporal donde el usuario puede hacer uso de internet para navegación, y obtener una buena experiencia el tiempo de conexión, las políticas de QoS para el Hotspot se basan en el control de ancho de banda y priorizaciones de tráfico. Cuando un usuario desea acceder a una página web a través del navegador está generando tráfico DNS es decir haciendo uso del protocolo UDP, para que la experiencia de usuario al consultar una nueva web no se vea ralentizada, se priorizó el trafico DNS sobre el tráfico generado en TCP que sería todo el tráfico generado al acceder a la web, además de estas dos políticas bajo los dos protocolos, existe la posibilidad de generación de tráfico de otro tipo al cual se lo denomino “RESTO”, con una prioridad menor incluso al tráfico WEB (TCP). Un punto importante es el protocolo ICMP, que para asegurar menores tiempos de respuesta se configuró la prioridad más elevada incluso sobre UDP.

CAPÍTULO II

2. MARCO METODOLÓGICO

Para la implementación de un sistema automatizado de administración de usuarios en un Hotspot a través de una API de Mikrotik se consideró una previa investigación acerca del lenguaje de programación necesario y las configuraciones para poder acceder al Router mediante la interfaz de programación. Este conocimiento previo permitió elaborar la interfaz de programación con los requerimientos de automatización necesarios para posteriormente seguir con las configuraciones en el Hotspot del control de ancho de banda para los usuarios.

2.1 Elementos de diseño en la implementación de la interfaz de programación API en Router Mikrotik

Uno de los principales objetivos es el de lograr la implementación de una interfaz de programación en Router Mikrotik, por lo que como previo se realizó el estudio necesario para poder llevar a cabo dicho objetivo.

El estudio llevara a cabo el análisis de código de programación utilizado para la API, la conexión del Router Mikrotik a la Interfaz de programación, y por ultimo todas las configuraciones necesarias a través de comandos para la automatización del manejo de usuarios en el Hotspot.

Teóricamente se definió las configuraciones por defecto para la API de Mikrotik, donde la opción esta deshabilitada y además de esto se encuentra definido el puerto de conexión 8728. Dicho puerto se debe tener a consideración ya que bajo ciertas circunstancias podría estar siendo ocupado, lo que conllevaría a tener que cambiar el número de puerto. Sin embargo, al cambiar el número de puerto en el dispositivo Mikrotik, se deberá cambiar de igual manera en la librería de la API para que no exista problemas de conexión.

A continuación, se puede observar el diagrama donde se establecen los pasos necesarios para el establecimiento de conexión entre API – Router. (Anexo A)

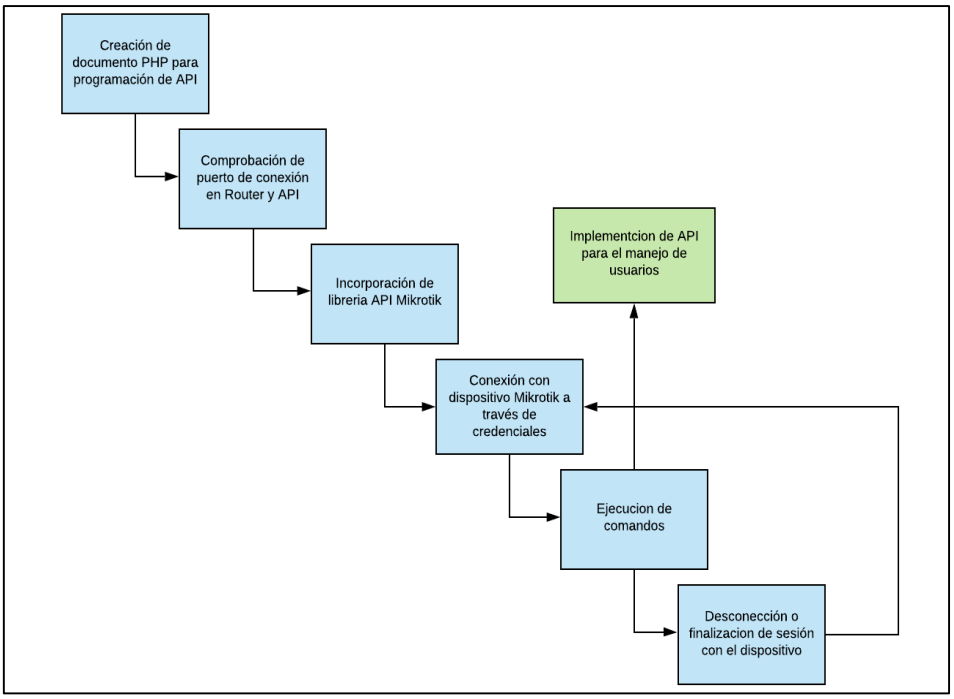


Figura 1-2: Diagrama de flujo de establecimiento de conexión con la API

Realizado por: Leon Daryel, 2021.

Cuando la conexión de haya conseguido satisfactoriamente se puede comprobar mediante la ejecución de comandos los mismos que serán enviados al dispositivo Mikrotik. No obstante, es necesaria la incorporación del servidor web para la ejecución.

Para el manejo de usuarios mediante la API fue necesaria la incorporación de funciones generadoras de credenciales aleatorias, con el propósito de evitar pérdidas de tiempo en la generación manual de cada credencial requerida. Básicamente hace uso de dos funciones, una para el nombre de usuario y la segunda para la contraseña, recalcando que en ambos casos el funcionamiento es el mismo.

El algoritmo aplicado para la generación de las funciones se puede observar a continuación detallada en la Figura. 2-2.

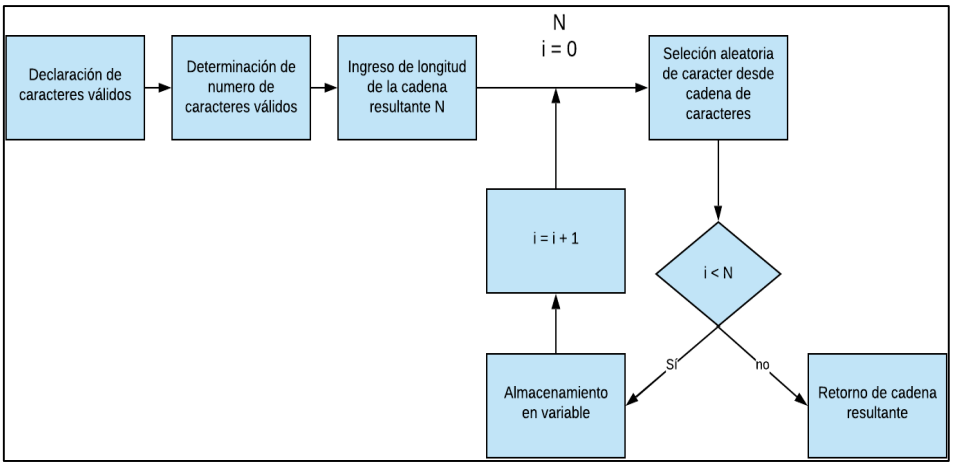


Figura 2-2: Algoritmo de función generadora de caracteres.

Realizado por: Leon Daryel, 2021.

Como se mencionó tanto para la obtención de la cadena con el nombre de usuario como para la cadena con la contraseña se aplica la misma lógica de algoritmo teniendo en cuenta que la variable final será lo que cambia.

2.2 Implementación y validación de instrumentos de prueba.

El proyecto incorpora materiales tanto de software como de hardware para la correcta ejecución.

2.2.1 Hardware.

Dentro de los componentes del Hardware utilizados para la implementación se puede distinguir lo siguiente:

Servidor: El servidor utilizado fue un computador portátil de la marca DELL con las siguientes especificaciones técnicas.

Tabla 1-2: Características Computador servidor

Parámetro	Valor
Modelo	DELL G5
Procesador	Intel Core i7-8750H
Frecuencia CPU	3.9 GHz
Memoria RAM	16 GB

Almacenamiento	1 TB
Sistema operativo	Windows 10
Arquitectura	64 bits

Fuente: Autor
Realizado por: Leon Daryel, 2021.

Servidor Hotspot: El dispositivo Mikrotik que se va a usar es el Routerboard rb951ui-2hnd que es un Router con capacidad de punto de acceso inalámbrico, las principales características técnicas del equipo (Anexo B), se detallan a continuación: (Tabla. 1-2).

Tabla 2-2: Características Routerboard rb951ui-2hnd

Parámetro	Valor
Arquitectura	MIPSBE
UPC	AR9344
Frecuencia CPU	600 MHz
Memoria RAM	128 MB
Almacenamiento	128 MB
Temperatura ambiente	-20 C + 50C
PoE en voltaje de entrada	9 – 30 V

Fuente: Mikrotik
Realizado por: Leon Daryel, 2021.

Este enrutador cuenta con antenas en su interior para funcionar como punto de acceso, bajo el estándar 802.11/b/g/n, en el canal de 2.4 GHz. Sin embargo, para poder cubrir una mayor extensión de territorio se podría hacer uno de una AP (Access Point) externo.

Proveedor: El servicio de internet es prestado por el proveedor CNT a través de enlace de fibra óptica que cuenta con el dispositivo enrutador de la marca Huawei con las siguientes características.

Tabla 3-2: Características enrutador de proveedor

Parámetro	Valor
Modelo	HG8245H
Puertos	1 interfaz GPON, 4Giga Ethernet, 2 POT, 1 USB, Wifi
Longitud de onda	Tx: 1310 nm, Rx: 1490 nm

WLAN	IEE 802.11 b/g/n
Rendimiento	Downstream 950 Mbps. Upstream 930 Mbps
Puerto de fibra	Downstream 2.5 Gbps. Upstream 1.25 Gbps
Potencia	< 8W

Fuente: Huawei

Realizado por: Leon Daryel, 2021

2.2.2 Software.

En la parte intangible como servidor web se utilizó Apache a través de la herramienta XAMPP con la versión 3.2.4.

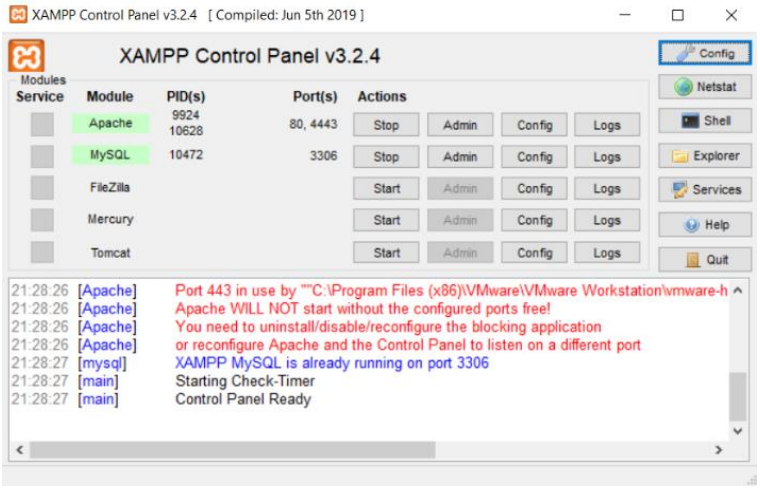


Figura 3-2: Servidor web a través de XAMPP

Realizado por: Leon Daryel, 2021

XAMPP, es un servidor disponible para sistema operativo Windows el cual permite entre otras opciones: Instalación y manejo de Apache, servidor y gestor de bases de datos, y por su puesto la característica a ser utilizada, el intérprete de PHP que va a ser el lenguaje de programación utilizado para la interfaz de programación.

PHP representa un lenguaje de programación de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Actualmente PHP representa una herramienta poderosa en el desarrollo de API´s, con un gran número de librerías disponibles puestas a disposición por parte de Mikrotik en colaboración de diferentes autores.

Tabla 4-2: Características PHP.

	PHP	DELPHI	C#	PYTHON	JAVA
Costo	✓	X	X	✓	X
Sintaxis	✓	X	X	✓	X
Seguridad	✓	-	-	✓	-
Popularidad	✓	X	X	X	X
Velocidad	✓	X	X	X	X

Fuente: Kinsta; Infranetworking; Python, 2021
Realizado por: Leon Daryel, 2021.

Tanto C Sharp, Delphi y Java son lenguajes de programación compilados mientras que para PHP y Python son lenguajes de programación interpretados, lo que requiere menos ya que correrá en casi cualquier servidor. Con respecto a la sintaxis tanto para PHP y Python de maneja tipado dinámico lo que favorece mucho al momento de comenzar con estos lenguajes, la comprobación de tipificación se realiza durante la ejecución y no durante la compilación. (Athanasias, 2014)

Teniendo en cuenta la seguridad cabe recalcar que tanto para PHP como para Python la conexión de la Api con el dispositivo se realiza a través del protocolo SSL y MD5, por defecto, en los demás casos la autenticación con SSL es adicional en una clase extra o por defecto no está disponible.

Con respecto a la utilización de uno u otro lenguaje, PHP represento el 79% en la utilización de páginas web hasta 2018, a pesar que en la actualidad a caído mucho dicho porcentaje sin embargo aún está sobre el 40% en la programación para sitios web. PHP en su versión 7.X es excepcionalmente rápido superando aun a Python según Benchmarks realizados por el proveedor de Wordpress hosting Kinsta. características orientadas a objetos que ofrece herencia o encapsulamiento de datos. (Ravoof, 2021)

Editor de texto: El editor de texto utilizado para la programación de la interfaz de programación fue Sublime Text el cual es un editor de código que entre los lenguajes de programación que soporta está el de PHP.

2.2.3 Desarrollo de escenarios y ambientes de prueba.

Para llevar a cabo el presente proyecto y con la finalidad de establecer escenarios comparativos para la realización de las pruebas de eficiencia en la administración de credenciales, se propuso

el primer escenario a través del uso de la herramienta Winbox, y el segundo que sería mediante la implementación de la API.

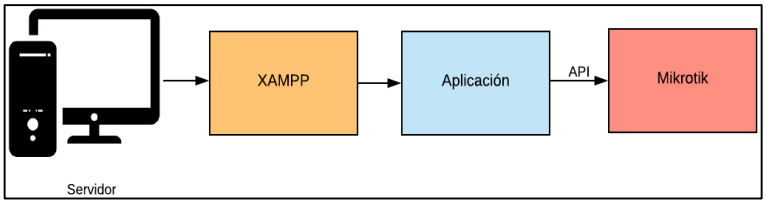


Figura 4-2: Escenario mediante API

Realizado por: Leon Daryel, 2021.

Accediendo a la aplicación codificada mediante el servidor XAMPP y obteniendo conectividad mediante al API al Router Mikrotik.

El indicador para obtener dicho objetivo fue el tiempo que toma la creación de un cierto número de usuarios en un ambiente frente al otro, consiguiendo así evaluar si existe una evidente mejoría en el tiempo al aplicar la automatización en el manejo de credenciales de usuario del Hotspot. (Anexo C).

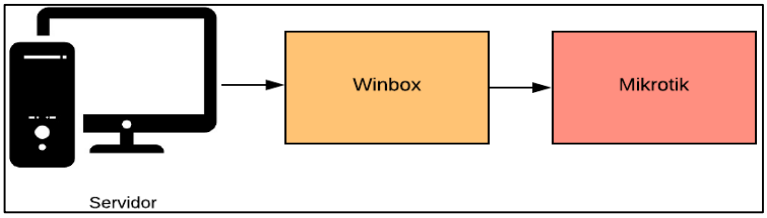


Figura 5-2: Escenario mediante de Winbox

Realizado por: Leon Daryel, 2021.

Winbox representa una utilidad que permite administrar dispositivos Mikrotik que dispongan como sistema operativo Router OS. Obteniendo accesibilidad al equipo por capa 2 y capa 3.

Ofrece un sistema de conexión basado en las credenciales del dispositivo al que se desea acceder. Las mismas que son Login, por defecto “admin”. Un Password de acceso el cual viene definido el blanco para la accesibilidad la primera vez. Y por último la dirección IP en caso de haberla configurado, sino lo que hace uso para la conexión es de la dirección MAC del dispositivo.

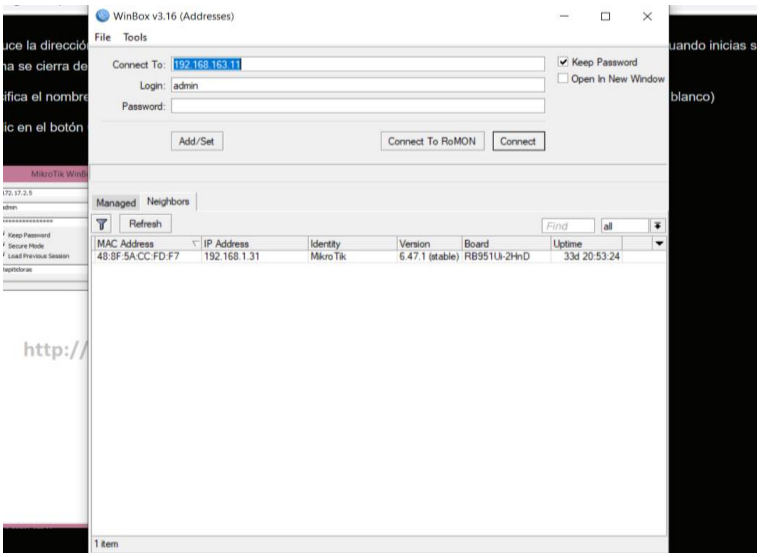


Figura 6-2: Acceso a Mikrotik mediante herramienta Winbox

Realizado por: Leon Daryel, 2021.

El segundo escenario propuesto para el presente trabajo realizado es el obtenido a través de la implementación de la API de Mikrotik. La misma se divide en dos partes, una parte funcional a través de código PHP, encargada de establecer la conexión y enviar los comandos al dispositivo Mikrotik, y la segunda parte, visual mediante código HTML, que es la interfaz gráfica con la finalidad de hacer más amigable la experiencia con el usuario.

Con respecto a los ambientes de prueba para le evaluación del rendimiento al aplicar políticas de calidad de servicio, se optó por dos ambientes, uno obteniendo los indicadores sin aplicar políticas de calidad de servicio, y el segundo ambiente que fue la obtención de los valores arrojados por los indicadores al aplicar las políticas de calidad de servicio. Los indicadores a ser evaluados se detallan en la Tabla. 4-2.

Tabla 5-2: Indicadores a evaluar

Parámetro	Unidad
Ancho de banda de descarga	Mbps
Ancho de banda de carga	Mbps
Latencia	ms
Jitter	ms
Pruebas ICMP	ms
Perdida de paquetes	-

Fuente: Autor

Realizado por: Leon Daryel, 2021.

Mencionados indicadores tienen su importancia en el análisis de tráfico y son considerados métricas por organizaciones reguladores de renombre como la IETF (Internet Engineering Task Force) y la ITU-T (International Telecommunication Union), establecidos bajo la recomendación ITU Y.1541.

2.2.4 Valoración de parámetros

En base a los parámetros antes expuestos se puede obtener una valoración mediante los valores establecidos bajo la Recomendación de ITU Y. 1541. Obteniendo valoraciones cualitativas en base a índices cuantitativos.

- **Latencia**

La recomendación establece un máximo para garantizar una buena calidad del servicio de 100 ms, cantidades sobre este valor muestran retardos muy elevados y con esto una disminución en la calidad del servicio prestado.

Tabla 6-2: Valoración de parámetro Latencia

Valoración	Cantidad (ms)
Sobresaliente	0 - 30
Muy buena	31 - 60
Buena	61 - 90
Regular	91 - 120
Pésima	121 - 150

Fuente: ITU Y. 1541.

Realizado por: Leon Daryel, 2021.

- **Jitter**

Lo establecido en la recomendación muestra un máximo de 50 ms para el parámetro Jitter, en base a esto se pudo obtener la escala de valoración para categorizar el parámetro de acuerdo a la métrica establecida.

Tabla 7-2: Valoración de parámetro Jitter

Valoración	Cantidad (ms)

Sobresaliente	0 - 10
Muy buena	11 -20
Buena	21 – 30
Regular	31 - 40
Pésima	41 - 50

Fuente: ITU Y. 1541.

Realizado por: Leon Daryel, 2021.

- **Perdida de paquetes**

Con respecto al parámetro perdida de paquetes la recomendación establece un máximo del 10% de perdida de paquetes en una trasmisión. En base a esto se pudo obtener la escala de valores para categorizar de acuerdo a la métrica expuesta.

Tabla 8-2: Valoración de parámetro Perdida de paquetes

Valoración	Cantidad (ms)
Sobresaliente	0 - 2
Muy buena	2 - 4
Buena	4 - 6
Regular	6 - 8
Pésima	8 - 10

Fuente: ITU Y. 1541.

Realizado por: Leon Daryel, 2021

2.3 Control de ancho de banda en usuarios de Hotspot

El control de ancho de banda hacia los usuarios del Hotspot se basa en un árbol de colas denominado “Queue Tree” por el cual se puede hacer una mejor distribución del recurso ancho de banda hacia los usuarios del Hotspot, el cual, una vez creado con perfiles pre configurados, y la conexión del Routerboard rb951ui-2hnd con la API. Se puede crear los “Queues” a través de la herramienta Winbox, en donde el tipo de cola o “Queue”, seleccionado para la implementación fue “PCQ” que es un encolamiento basado en conexión.

El límite máximo del tamaño de la cola establecido para efectos del presente trabajo fue de 2M de manera simétrica, en base a las limitaciones del proveedor de internet con el que se realizó las pruebas. A partir de dicha cantidad se estableció que los perfiles para los usuarios que deseen contratar el servicio temporal va a contar con un máximo de ancho de banda de bajada de 0.5Mbps

de igual manera simétrica, con el fin que exista el control de ancho de banda y no exista casos donde un solo usuario abarque todo el recurso.

Dicho valor de asignación al ancho de banda se tomó en base a las pautas provistas por la Comisión Federal de Comunicaciones (FCC), en su guía de velocidades de ancho de banda (Anexo D), donde se establece valores de anchos de banda mínimos según las necesidades o actividades que se van a llevar a cabo.

Tabla 9-2: Guía de velocidades de ancho de banda (FCC)

Actividad	Velocidad mínima (Mbps)
Navegación general y Email	0.5
Streaming de audio	0.5
Voz sobre IP	0.5
Estudiantil	5-25
Telecomunicaciones	5-25
Redes Sociales	1

Fuente: Comisión Federal de comunicaciones, 2018

Realizado por: Leon Daryel, 2021

Como actividad de referencia se estableció la navegación en general y manejo de correos electrónicos, estableciendo así un mínimo de 0.5 Mbps, cabe recalcar que la evaluación se la llevaría por tanto en el peor escenario posible, es decir bajo el mínimo del recurso, de esta manera, si se consigue bajo estas circunstancias, por consiguiente, en escenarios con mayor velocidad quedarían comprobados.

Con respecto a las colas (QUEUES) generadas, en la Figura. 5.2. Se puede observar la tabla resumen del árbol de colas generado donde se especifica tanto para descarga como para subida.

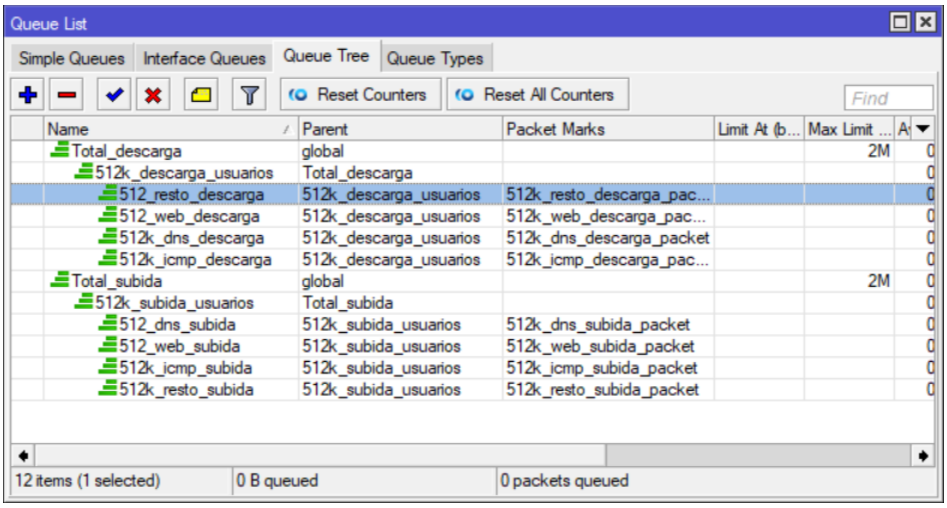


Figura 7-2: Árbol de colas generado

Realizado por: Leon Daryel, 2021.

Del árbol generado se puede deducir los valores esperados al momento de realizar las pruebas, garantizando mediante la creación de las colas el valor mínimo al que cada usuario está accediendo. Con la finalidad de establecer más adelante las políticas de calidad de servicio, en las colas hijas se ha especificado 4 tipos de tráfico que se podría generar en el Hotspot por los usuarios.

2.3.1 Políticas de calidad de servicio.

A partir de las colas generadas anteriormente, y siguiendo la teoría al momento de aplicar calidad servicio, detallada en el capítulo previo, el siguiente paso fundamental para ver implementado el árbol de colas, es la marcación de paquetes y conexiones. Dicho proceso conlleva relación cercana al árbol generado, ya que se realizará la marcación según el tipo de trafico al cual las colas se generaron, en base al valor de ancho de banda que se pre configuró para los perfiles de los usuarios, es decir ancho de banda destinado a Navegación en general y Email, se puede deducir el tráfico principal al cual accederían dichos usuarios, teniendo en cuenta esto, el trafico sobre el que se llevó control se detalla en la Tabla. 5.2.

Tabla 10-2: Tipo de tráfico considerado para las políticas de calidad de servicio

Tráfico	Protocolo	Puerto
DNS	UDP	53
ICMP	ICMP	-
WEB	TCP	80/443
Resto	-	-

Fuente: Autor

Realizado por: Leon Daryel, 2021.

En base a dichas consideración a manera de resumen de las políticas de calidad de servicio generadas, la Figura. 6.2. muestra a manera de resumen el Mangle general obtenido (Anexo E), detallando, además, el Mangle de Forward, Mangle de Prerouting, Mangle de Input y por ultimo Mangle de Output. Detallado teóricamente para llevar a cabo la marcación completa de paquetes y conexiones

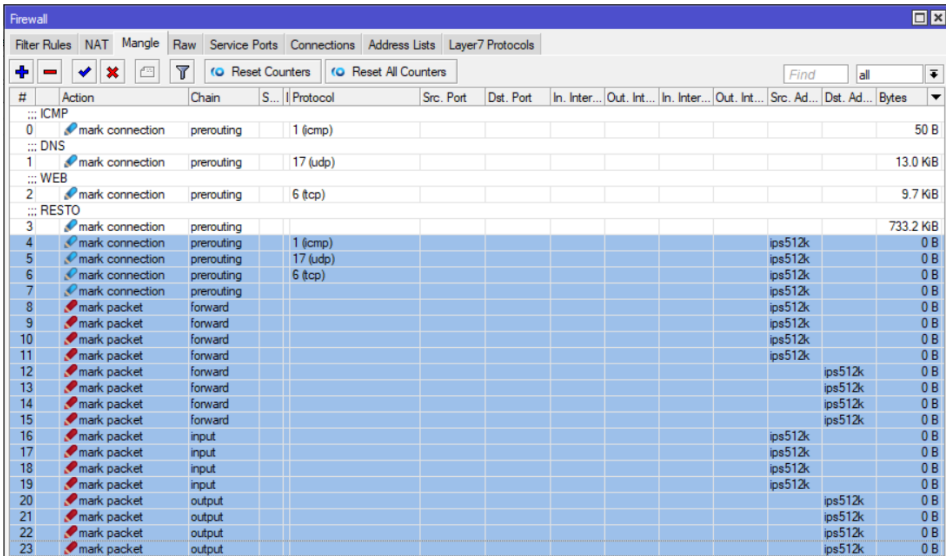


Figura 6-2: Resumen Mangle de marcación de paquetes y conexiones

Realizado por: Leon Daryel, 2021.

2.4 Técnica de recolección de datos

A continuación, se detalla las técnicas de recolección de datos para cada uno de los parámetros que se van a evaluar, tanto para la interfaz de creación automática de credenciales de usuario, como para los indicadores de la calidad de servicio al aplicar el control de ancho de banda con las políticas de calidad de servicio.

2.4.1 Eficiencia en la administración de credenciales de usuario

Teniendo en cuenta los dos escenarios presentados, las pruebas para la evaluación de la interfaz gráfica de creación automática de credenciales de usuario a través de la API se realiza mediante el indicador tiempo, el mismo que evidenciara la mejoría de ser el caso, de la creación de usuarios a través de la API. La evaluacion consto de 10 pruebas en donde en cada una se llevó a cabo la

creación de 10 usuarios, llegando a un tope de 100 usuarios creados al final de la evaluación. Obteniendo así el tiempo que tomaría la creación manual de 10 usuarios (primer escenario), y cuanto tomaría la creación de 10 usuarios mediante la API (segundo escenario). De esta manera en el primer escenario al cabo de las 10 pruebas pudiendo obtener una media de creación de 10 usuarios para poder evidenciar el comportamiento con respecto a la creación de distintos números de usuarios. A través de la aplicación de la siguiente formula:

$$Tiempo\ de\ creacion\ de\ 1\ usuario = \frac{Tiempo\ de\ prueba}{10}$$

Se puede obtener el valor del tiempo de creación de un usuario, teniendo en cuenta las 10 pruebas realizadas se llegaría al promedio del mismo, con esto pudiendo reflejar para N usuarios requeridos a través de la fórmula:

$$Tiempo\ de\ creacion\ de\ N\ usuarios = Tiempo\ de\ creacion\ de\ 1\ usuario * N$$

Donde N representa el número de usuarios que se creó. Pudiendo evidenciar así el comportamiento en cualquier caso de número de usuarios a crear,

Con respecto al segundo escenario teniendo en cuenta que el número de usuarios a crear no afectara el tiempo, su comportamiento es plano a lo largo de la creación de 1 a 100 usuarios de la evaluación. De esta manera de las 10 pruebas realizadas el valor promedio de la creación de 10 usuarios se refleja a cualquier número N de usuarios a crear.

2.4.2 Evidencia del control mesurado de ancho de banda a los usuarios

Las pruebas de la evidencia de una suministración controlada del recurso ancho de banda a los usuarios, se llevó a cabo a través de la obtención del valor otorgado del recurso sin aplicar un control de ancho de banda y segundo, aplicando un control de ancho de banda. De esta manera mediante la toma de datos mediante 10 pruebas, se pudo obtener el valor promedio. Y en base al mismo mediante la fórmula de desviación estándar:

$$\sigma = \sqrt{\frac{\sum_i^N (X_i - \bar{X})^2}{N}}$$

Se puede obtener la variación en el conjunto de datos, llegando a evidenciar que, a menor valor de desviación, el valor entregado del recurso está más cercano a un valor particular. Con respecto al segundo escenario este valor en particular debería rondar valores cercanos a 0.5 Mbps, con su respectiva desviación de dicho valor. Esperándose un comportamiento casi plano entorno al valor pre configurado.

2.4.3 Evaluación de parámetros de calidad de servicio

Para la recolección de datos de los parámetros Jitter y latencia se lo realizo a través del navegador de un dispositivo conectado a la red del Hotspot, mediante la herramienta de Speedtest.net. cuyo test muestra valores de ancho de banda de subida y bajada, ping o latencia y j Jitter o fluctuación. Con las 10 pruebas realizadas se puede llegar a un promedio de los valores de dichos parámetros en la red, para posteriormente comparar en ambos ambientes de pruebas.

Con respecto a la perdida de paquetes, se realizó pruebas de ICMP generando paquetes en este caso 20 paquetes, bajo un consumo de ancho de banda para evidenciar que las políticas de calidad de servicio estén surtiendo efecto. Por lo que se tendría los datos necesarios para la obtención del porcentaje de perdida de paquetes detallado en la siguiente fórmula.

$$PI\ (\%) = \frac{Penviados - Precibidos}{Penviados} * 100$$

El tráfico generado de los 20 paquetes se lo realizo a través de la herramienta Winbox accediendo al protocolo ARP realizando Ping al dispositivo autenticado en la red del Hotspot.

CAPÍTULO III

3. MARCO DE RESULTADOS

3.1 Resultados del manejo de usuarios mediante API.

Con la implementación de una interfaz visual el proceso se hace más intuitivo para el usuario administrador. A través de menús y espacios donde pueda ingresar la información que requiere generar, simplemente accediendo al menú generado en la API, el administrador puede ingresar lo parámetros que desee, como seleccionar un perfil de los pre establecidos, seleccionar la longitud que tendrán los nombres de los usuarios y contraseñas, ingresar un comentario que es el caso del presente trabajo se trabajó como el valor que será cobrado al cliente, el tiempo se le otorgara al cliente que va sujeto al perfil al cual se está estableciendo, y por el numero el número de usuarios que desea crear de una sola vez, este parámetro es el que mayor eficiencia provoca en el proceso ya que el administrador de la red puede genere de una sola vez y ejecutando un solo proceso el número de usuarios que desee, los tiempos que toma la generación manual de credenciales se ven minimizados exponencialmente, para mostrar este resultado se siguió con las pruebas detallas en el previo capitulo para la comprobación de la eficacia en la administración de credenciales.

Tabla 1-3: Pruebas de deducción de tiempos para la creación manual de credenciales de usuarios.

Prueba	Valor
1	268.8 s
2	197.1 s
3	191.2 s
4	198.9 s
5	186.7 s
6	182.8 s
7	186.1 s
8	199.8 s
9	182.1 s
10	179.8 s
Promedio	197.33 s

Fuente: Autor

Realizado por: Leon Daryel, 2021.

Prosiguiendo con la prueba se obtuvo los tiempos mediante la utilización de la interfaz (API) detallados en la Tabla. 4-3.

Tabla 2-3: Pruebas de deducción de tiempos para la creación automática de credenciales de usuarios.

Prueba	Valor
1	12.72 s
2	12.49 s
3	13.02 s
4	11.98 s
5	12.56 s
6	12.71 s
7	12.29 s
8	12.48 s
9	13.10 s
10	12.65 s
Promedio	12.602 s

Fuente: Autor

Realizado por: Leon Daryel, 2021.

En la gráfica a continuación se puede observar de manera comparativa ambos resultados de tiempos en segundos en ambos métodos.

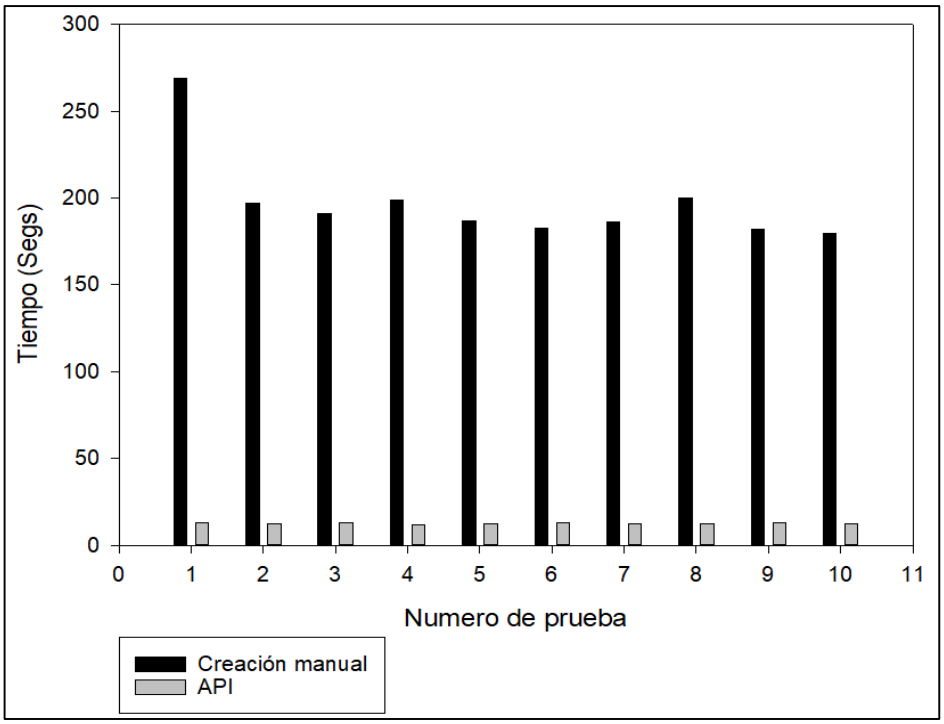


Gráfico 1-3: Tabla de resumen de tiempos obtenidos en cada método

Realizado por: Leon Daryel. 2021.

Como se puede observar en la Gráfica 1-3, la diferencia de tiempos que toma la creación de 10 usuarios de manera manual contra la interfaz gráfica mediante API es considerable, y a partir de las tablas previas se puede obtener que la diferencia entre los promedios de ambos métodos después de 10 pruebas, es de 184.728 segundos, este resultado tiene un comportamiento lineal ya que a medida que el número de usuarios a crear aumente, mayor será la diferencia de tiempo entre el método manual y el propuesto en el presente trabajo, por lo que se puede deducir una notable mejoría en cuanto al tiempo empleado para la creación de credenciales de usuario además que a medida que el número de usuarios requeridos aumente, mayor será el beneficio obtenido de crearlos mediante la API a través de la interfaz gráfica. Dicho comportamiento se puede observar en la Gráfica 2-3, que muestra el resultado de la diferencia de tiempos que toma entre el primer método y el segundo.

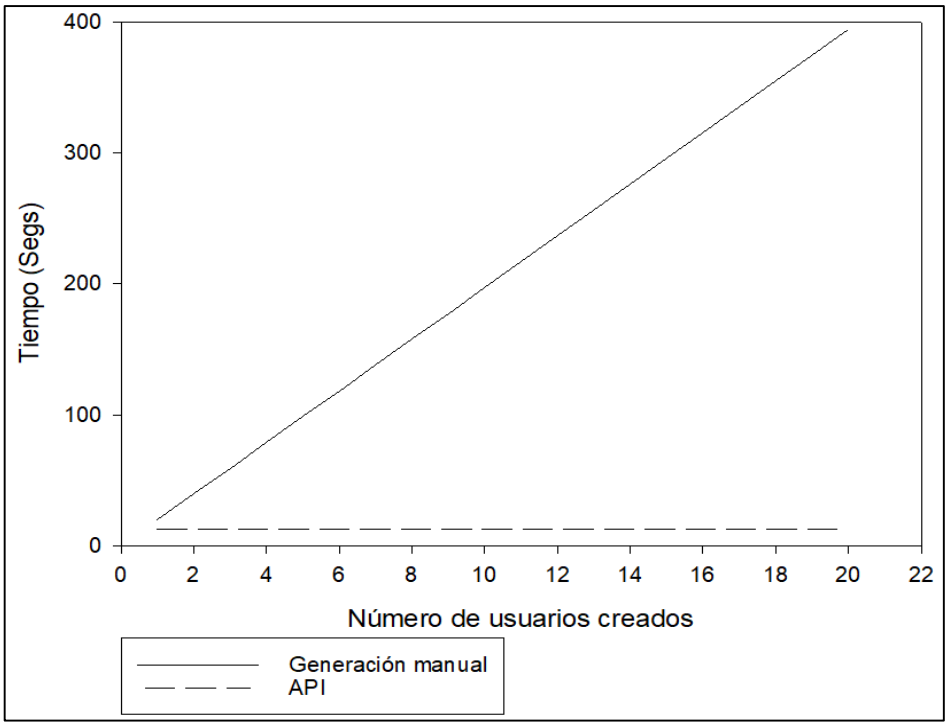


Grafico 2-3: Diferencias de tiempos de creación de usuarios entre ambos métodos

Realizado por: Leon Daryel, 2021.

3.3 Evaluación del control mesurado de ancho de banda

En la Tabla 5-3. Se puede observar el análisis realizado al parámetro ancho de banda de descarga en el primer ambiente de pruebas es decir sin la aplicación de políticas de calidad de servicio.

Tabla 3-3: Pruebas de ancho de banda de descarga sin aplicar políticas QoS.

Prueba	Valor
1	14.19 Mbps
2	9.92 Mbps
3	13.27 Mbps
4	16.69 Mbps
5	14.47 Mbps
6	18.74 Mbps
7	16.69 Mbps
8	13.70 Mbps
9	9.92 Mbps
10	12.02 Mbps

Promedio	13.961 Mbps

Fuente: Autor

Realizado por: Leon Daryel, 2021.

A través de la fórmula de la desviación estándar detallada en el previo capitulo se puede obtener el valor de la variación del ancho de banda que un usuario obtendría si no se hiciera un control de dicho recurso. Aplicando mencionada formula al grupo de datos obtenidos de valores de anchos de banda de la Tabla 5-3.

$$\sigma = 2.7389$$

Con respecto al parámetro ancho de banda de carga se obtuvo los siguientes valores en el conjunto de pruebas realizadas.

Tabla 4-3: Pruebas de ancho de banda de carga sin aplicar políticas QoS.

Prueba	Valor
1	11.91 Mbps
2	16.89 Mbps
3	14.54 Mbps
4	17.24 Mbps
5	18.68 Mbps
6	17.27 Mbps
7	12.98 Mbps
8	12.17 Mbps
9	14.53 Mbps
10	13.14 Mbps
Promedio	14.935 Mbps

Fuente: Autor

Realizado por: Leon Daryel, 2021.

Y una vez aplicada la fórmula de la desviación estándar detallada anteriormente, se obtuvo el siguiente valor.

$$\sigma = 2.2972$$

Analizando el segundo ambiente de pruebas, es decir aplicando las políticas de calidad de servicio se obtuvo los datos recolectados en la Tabla. 9-3.

Tabla 7-3: Pruebas de ancho de banda de descarga aplicando políticas QoS

Prueba	Valor
1	0.52 Mbps
2	0.5 Mbps
3	0.52 Mbps
4	0.53 Mbps
5	0.53 Mbps
6	0.52 Mbps
7	0.51 Mbps
8	0.5 Mbps
9	0.51 Mbps
10	0.53 Mbps
Promedio	0.517 Mbps

Fuente: Autor

Realizado por: Leon Daryel, 2021.

De igual manera se procedió a la recolección de datos de ancho de banda de carga en la red aplicando las políticas de QoS. Los datos de las 10 pruebas se resumen en la Tabla 10-3.

Tabla 8-3: Pruebas de ancho de banda de carga aplicando políticas QoS

Prueba	Valor
1	0.57 Mbps
2	0.52 Mbps
3	0.54 Mbps
4	0.55 Mbps
5	0.52 Mbps
6	0.52 Mbps
7	0.54 Mbps
8	0.52 Mbps
9	0.52 Mbps
10	0.53 Mbps
Promedio	0.533 Mbps

Fuente: Autor

Realizado por: Leon Daryel, 2021.

Y en base a los datos registrados de anchos de banda de subida y de bajada, para la desviación estándar del parámetro ancho de banda de descarga, se obtuvo el siguiente valor.

$$\sigma = 0.011$$

Mientras que, para el parámetro ancho de banda de carga, el valor de desviación estándar resulto el siguiente.

$$\sigma = 0.016$$

El valor promedio otorgado de ancho de banda de banda a los usuarios del Hotspot fue de 0.517 para descarga y de 0.533 para carga; valores esperados del aplicar el control, al realizar la comparación de desviación estándar, se obtiene para el método manual sin aplicar políticas de calidad de servicio, un valor elevado con respecto al método mediante interfaz gráfica con API, aplicando políticas de calidad de servicio, lo que se puede deducir que sin un control de ancho de banda, al usuario no se puede asegurar un valor de ancho de banda, ya que las variaciones de este recurso otorgado estaría en constante cambio, mientras que de aplicar un control de ancho de banda, la desviación estándar disminuye considerablemente, indicador el cual permite deducir que el usuario recibiría un ancho de banda más estable, acorde al valor establecido pre configurado.

Este comportamiento se puede ver expresado en Gráfica 3-3 donde el comportamiento para el escenario con un control de ancho de banda se observa más lineal en comparación a lo que se puede observar sin aplicar un control de ancho de banda.

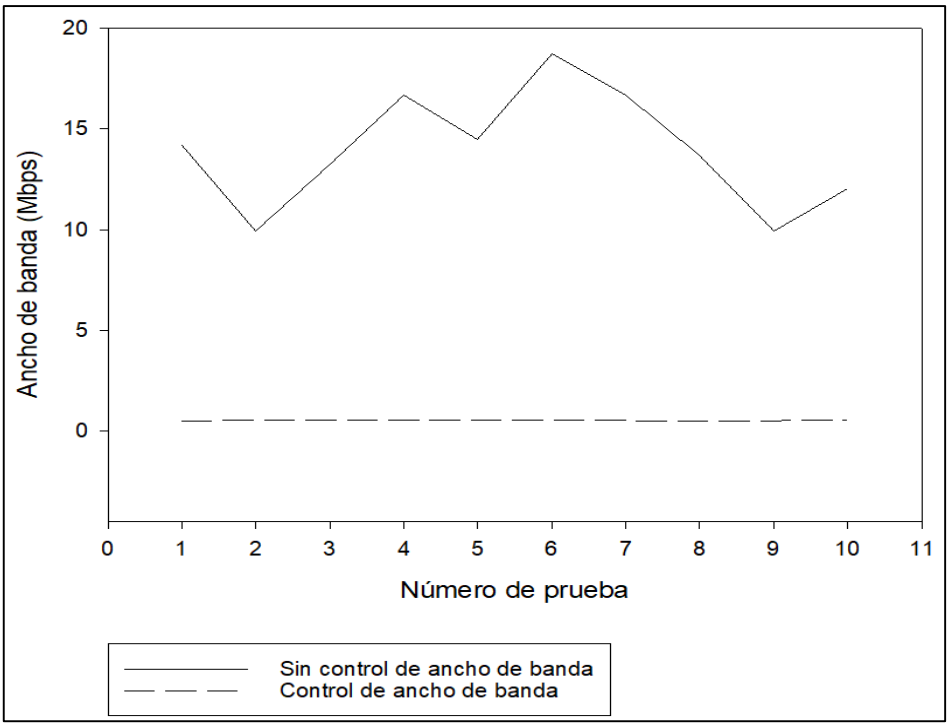


Grafico 3-3: Comportamiento de ancho de banda con y sin aplicar control.

Realizado por: Leon Daryel, 2021

3.4 Evaluación de los parámetros de calidad de servicio.

Analizando el siguiente parámetro, latencia, en las 10 pruebas realizadas se obtuvieron los siguientes datos detallados en la Tabla 9-3.

Tabla 7-3: Pruebas de latencia sin aplicar políticas QoS

Prueba	Valor
1	10 ms
2	10 ms
3	9 ms
4	9 ms
5	11 ms
6	9 ms
7	8 ms
8	9 ms
9	9 ms
10	12 ms

Promedio	9.6 ms

Fuente: Autor

Realizado por: Leon Daryel, 2021

Como siguiente se recolecto los datos obtenidos del parámetro "Jitter", la tabla a continuación muestra valores típicos durante las 10 pruebas realizadas.

Tabla 8-3: Pruebas de "Jitter" sin aplicar políticas QoS

Prueba	Valor
1	3.68 ms
2	22.61 ms
3	8.90 ms
4	17.68 ms
5	14.49 ms
6	17.53 ms
7	17.36 ms
8	9.27 ms
9	30.43 ms
10	14.63 ms
Promedio	15.658 ms

Fuente: Autor

Realizado por: Leon Daryel, 2021

Por otra parte, analizando el segundo ambiente el valor de latencia de igual manera se presentan los datos obtenidos en la tabla a continuación.

Tabla 9-3: Pruebas de latencia aplicando políticas QoS

Prueba	Valor
1	9 ms
2	13 ms
3	10 ms
4	13 ms
5	8 ms
6	10 ms
7	10 ms
8	8 ms

9	9 ms
10	9 ms
Promedio	9.9 ms

Fuente: Autor

Realizado por: Leon Daryel, 2021

Y como último parámetro de análisis para la comparación están las pruebas del Jitter al aplicar las políticas de calidad de servicio y control del ancho de banda.

Tabla 12-3: Pruebas de "Jitter" aplicando políticas QoS

Prueba	Valor
1	54.68 ms
2	32.4 ms
3	6.07 ms
4	11.02 ms
5	2.42 ms
6	5.96 ms
7	11.51 ms
8	1.85 ms
9	1.84 ms
10	9.21 ms
Promedio	13.696 ms

Fuente: Autor

Realizado por: Leon Daryel, 2021

Con esto se ha podido realizar un análisis de los parámetros de la red para poder efectuar la comparación al aplicar las políticas de QoS en el Hotspot.

Una vez concluidas las pruebas de evaluación de la red a través del análisis de los parámetros, se pudo llevar a cabo la comparación punto por punto donde como primer objetivo de dicha comparación estuvo el parámetro Latencia, se pudo observar que se mantiene dentro de los parámetros para ser considerada Sobresaliente. Mostrando según las pruebas realizadas, que en la red aplicando el control de ancho de banda con políticas de calidad, el valor en promedio de latencia medida en ms, casi se mantiene inmutable, plasmado en la Gráfica. 4-3.

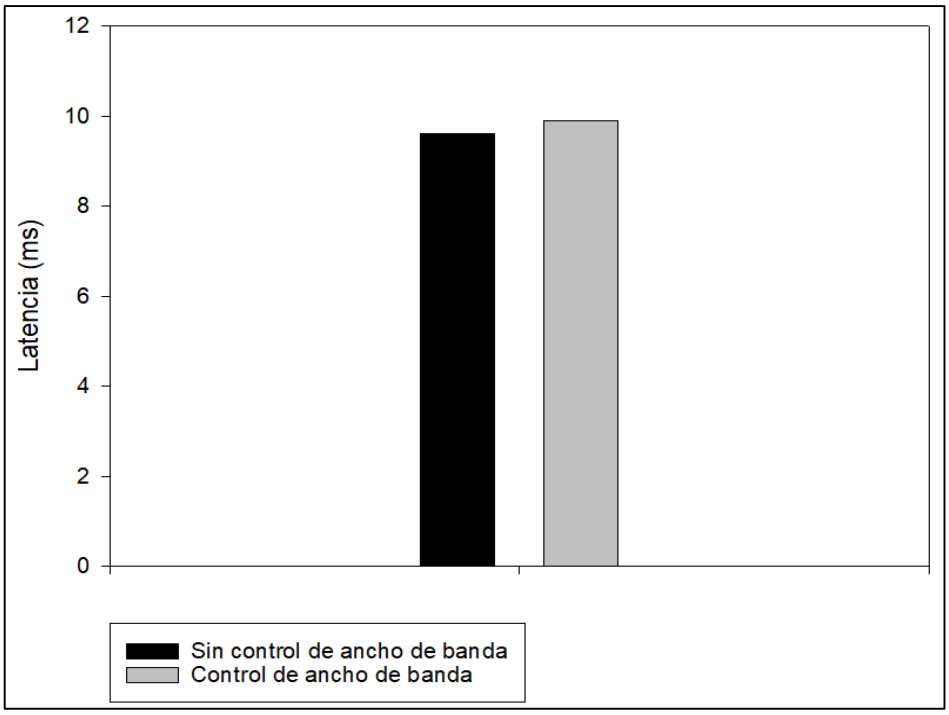


Grafico 4-3: Latencia promedio con y sin aplicar control de ancho de banda

Realizado por: Leon Daryel, 2021

Por último, el parámetro que se comparo fue el "Jitter", donde el comportamiento apreciado con respecto a la latencia se mantiene, se puede corroborar una disminución al aplicar el control de ancho de banda con políticas de calidad de servicio, dicho comportamiento se puede observar en la Gráfica 5-3.

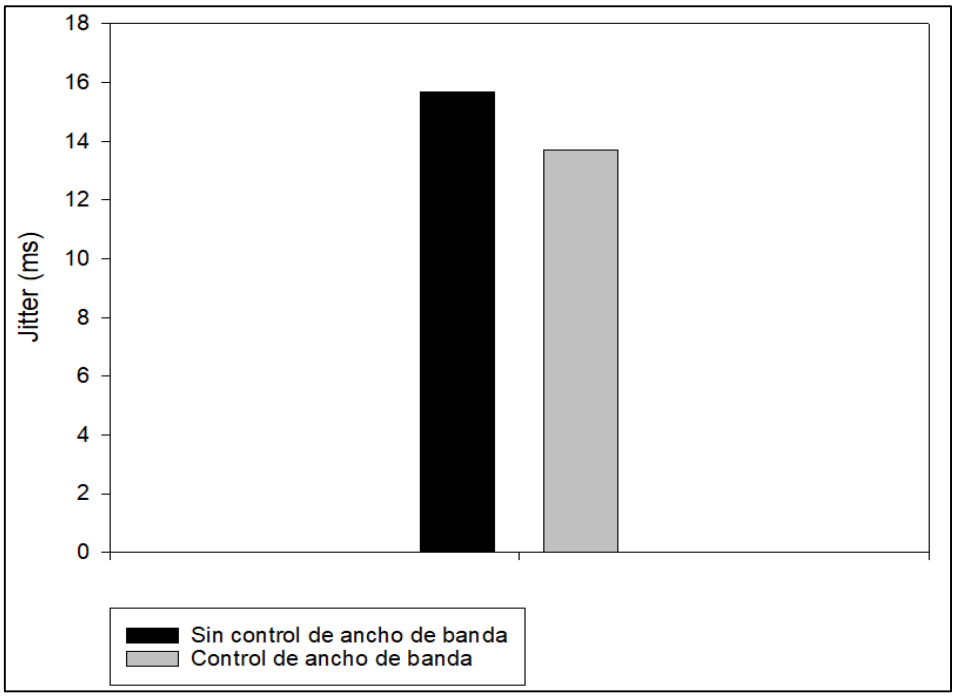


Grafico 5-3: Jitter promedio medido en ms con y sin aplicar control de ancho de banda

Realizado por: Leon Daryel, 2021

Viendo plasmado en los resultados, la configuración realizada para el control de ancho de banda con políticas de calidad.

3.5 Pruebas ICMP y perdida de paquetes.

Mediante las pruebas detalladas en el previo capitulo se pudo obtener los valores de porcentaje de perdida de paquetes a través de las pruebas de ICMP.

Tabla 13-3: Pruebas de ICMP sin aplicar políticas QoS

Prueba	Valor
1	352 ms
2	132 ms
3	172 ms
4	153 ms
5	336 ms
6	-
7	-
8	-
9	-
10	-
11	-
12	300 ms
13	120 ms
14	375 ms
15	715 ms
16	412 ms
17	114 ms
18	322 ms
19	473 ms
20	654 ms

Fuente: Autor

Realizado por: Leon Daryel, 2021

A partir de la tabla se puede observar como existe un porcentaje de perdida de paquetes, en base a la fórmula del porcentaje de perdida de paquetes.

Los 6 paquetes sin respuesta representan un porcentaje del 30% del total de 20 paquetes enviados durante las pruebas de ICMP. Además de un promedio de tiempo de envío de paquetes de 329.08 ms

De igual manera las pruebas de ICMP realizadas en la red, aplicando las políticas de calidad, arrojo los valores recolectados en la tabla a continuación.

Tabla 12-3: Pruebas de ICMP aplicando políticas QoS

Prueba	Valor
1	46 ms
2	1 ms
3	32 ms
4	35 ms
5	1 ms
6	1 ms
7	41 ms
8	0 ms
9	1 ms
10	1 ms
11	2 ms
12	23 ms
13	1 ms
14	2 ms
15	23 ms
16	1 ms
17	2 ms
18	0 ms
19	1 ms
20	0 ms

Fuente: Autor

Realizado por: Leon Daryel, 2021

Llevando a cabo un breve análisis de la Tabla 14-3. Se puede observar que no existe perdida de paquetes, además que, en los tiempos de envió de paquetes se puede observar una considerable

disminución, con un promedio de envío de paquetes de 10.7 ms. Como resultado de la aplicación de políticas de calidad y la priorización de tráficos. La Gráfica 6-3 muestra la representación del promedio de los datos obtenidos de las pruebas Ping.

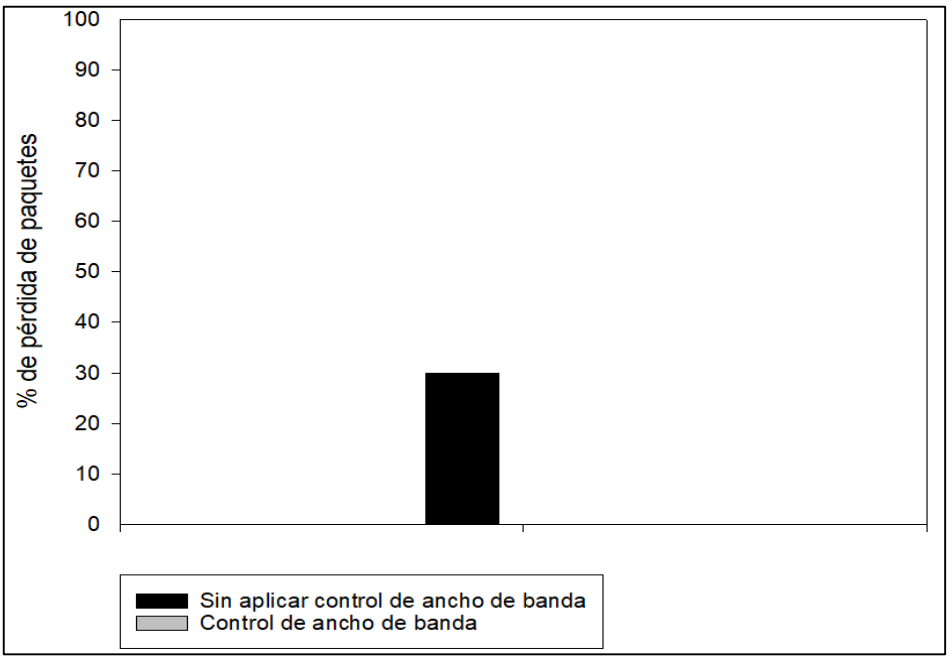


Grafico 6-3: Porcentaje de pérdida de paquetes en pruebas ICMP

Realizado por: Leon Daryel, 2021

3.6 Tabla comparativa resumen

Tabla 15-3: Tabla comparativa entre los dos escenarios.

Parámetro / Escenario	Sin Control de AB y QoS	Control de AB y QoS
Ancho de banda de descarga	13.961 Mbps	0.517 Mbps
Ancho de banda de carga	14.935 Mbps	0.53 Mbps
Latencia	9.6 ms	9.9 ms
Jitter	15.658 ms	13.696 ms
Perdida de paquetes (%)	30%	0%

Fuente: Autor

Realizado por: Leon Daryel, 2021

Una vez obtenida la tabla final de resultados de pruebas realizar en la red, se puede visualizar las diferencias claras en el escenario sin aplicar control de ancho de banda con políticas de calidad,

frente al escenario con control de ancho de banda con políticas de calidad, los parámetros de latencia en el segundo escenario se mantuvieron dentro del rango Sobresaliente, de igual manera con el Jitter. Un parámetro importante fue la priorización del tipo de trafico ICMP, siguiendo la conceptualización de priorización de trafico tratada en el capítulo 2, para el escenario planteado en el trabajo se obtiene una disminución del tiempo de envío de paquetes y además de erradicar a 0 el porcentaje de perdida de paquetes durante las pruebas realizadas.

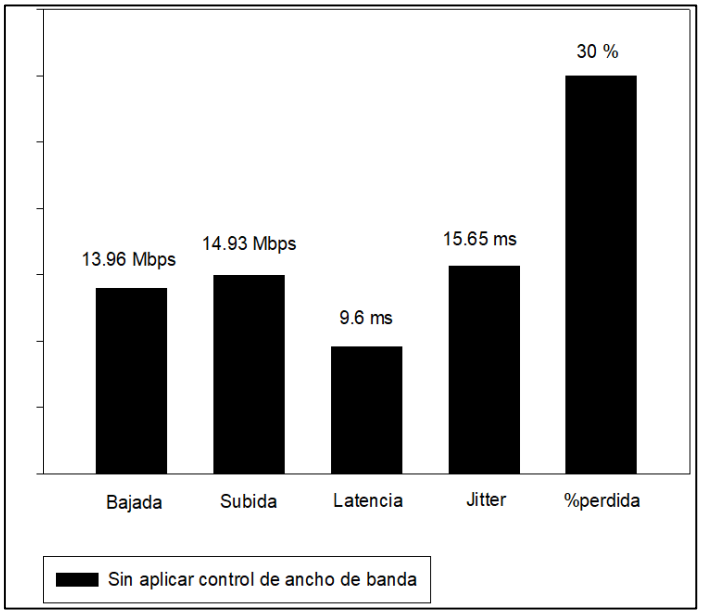


Grafico 7-3: Ambiente sin control de ancho de banda

Realizado por: Leon Daryel, 2021

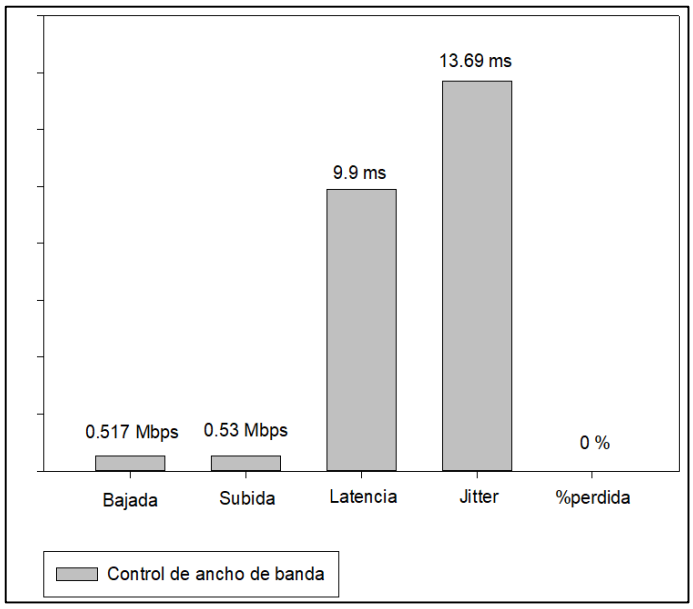


Grafico 8-3: Ambiente con control de ancho de banda

Realizado por: Leon Daryel, 2021

CONCLUSIONES

- Para la red Hotspot, se estableció las políticas de control de ancho de banda y prioridades al tráfico que más comúnmente generarían los usuarios y al cual se enfoca este tipo de red (Navegación general y Email), es decir peticiones DNS, (puerto 53) y trafico WEB puerto (80 para HTTP y 443 para HTTPS).

- Se estableció exitosamente la conexión de la API de Mikrotik con el Router implementando así una función generadora de caracteres aleatorios mediante lenguaje PHP para la generación de credenciales (nombres y contraseñas) de usuarios incorporando además la interfaz gráfica para manejo más intuitivo de las misma

- Al evaluar el rendimiento de la aplicación de políticas Queue Tree en los parámetros de evaluación tales como Latencia manteniéndose con una valoración sobresaliente, Jitter con un aproximado de 10%. Y por último un porcentaje de 96% de mejora en los tiempos de respuesta de pruebas PING con una tasa de 0% de perdida de paquetes en la red al presentar una distribución equitativa del recurso ancho de banda.

- Al implementar la API para el control de usuarios con control de ancho de banda, se pudo observar una mejora cerca del 94% en el tiempo de la generación de 10 credenciales de usuario realizadas para el Hotspot, viéndose este porcentaje elevado linealmente a medida que el requerimiento de credenciales aumenta.

RECOMENDACIONES

- Se recomienda deshabilitar las interfaces virtuales generadas por otros programas o a su vez cambiar el puerto del módulo APACHE al momento de inicializar el servidor XAMPP ya que se puede producir conflictos de puertos.

- Dependiendo de la calidad de servicio que se desea otorgar a los usuarios del Hotspot, se recomienda distribuir a cada usuario un mayor ancho de banda ya que el usado en el presente trabajo resulto justo al momento de visualizar videos u otro contenido multimedia.

- Se recomienda tener conocimiento de lenguaje de programación PHP y HTML que son los lenguajes usados para el manejo de la API de Mikrotik para poder implementar más funciones acordes a las necesidades o peticiones que el administrador del Hotspot llegara a tener.

- Se recomienda manejar la clase API de "Routeros" actualizada, ya que versiones anteriores a la actual presenta conflictos al momento de conectarse con versiones más recientes de routers Mikrotik

- Para ambientes más amplios o para cubrir zonas de mayor extensión al aire libre se recomienda hacer uso de antenas de puntos de acceso, ya que la interfaz inalámbrica del router Mikrotik podría no ser suficiente para cubrir toda la zona, además que no presenta protección frente a agua ni polvo para estar al aire libre.

- Se recomienda para futuros trabajos tener en cuenta las características de multivendor en una API para comprobar los resultados en diferentes equipos enrutadores.

BIBLIOGRAFIA

Andreu, J. *Redes inalámbricas (Servicios en red)*. Lugar, España: Editex, 2011, pp. 211 – 214.

Pedron, Carlos. "Cisco Routers vs MikroTik Routers and Switches". "TrustRadius", [en línea], 2018, Colombia, pp. 44 – 51. [Consulta: 24 marzo 2021]. Disponible en: https://www.trustradius.com/users/5b0fff35193eb40017805b10.

Arias, M. A. *Introducción a PHP*. Lugar, España: IT Campus Academy, 2013, pp 13 – 18.

Barnes, N. *API PHP class* [en línea] - MikroTik Wiki, 2015. [Consulta: 2 enero 2021]. Disponible en: https://wiki.mikrotik.com/wiki/ API_PHP_class.

Caiza, F & Lenin. J. Estudio comparativo de la implementación de un portal cautivo mediante las tecnologías Mikrotik y Cisco para mejorar el rendimiento de una red inalámbrica en MiPyMES. (Tesis de pregrado). Escuela Superior Politécnica de Chimborazo, Riobamba. 2018. pp. 40 – 48.

Clep, Mario. "Calidad de Servicio. Implementación de un Sistema de QoS en RouterOS". "Mikrotik". [en línea], 2009, United States of America, pp. 7 – 15. [Consulta: 18 enero 2021]. Disponible en: https://mum.mikrotik.com//presentations/AR09/mario_clep.pdf

Cruz Felipe, M. D. R., Martínez Gómez, R., & Crespo García Y. "Análisis de la QoS en redes inalámbricas". "Revista Cubana de Ciencias Informáticas", [en línea], 2013, Cuba, pp. 86 – 96. [Consulta: 4 febrero 2021]. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992013000100010.

Diaz, Juan. "HTB y herramientas de medida". "Mikrotik", [en línea], 2008, United States of America, pp. 10 – 21. [Consulta: 8 febrero 2021]. Disponible en: https://mum.mikrotik.com/presentations/BO19/presentation_7233_1575588079.pdf

Dobladez, Maximiliano. "Administración y control de ancho de banda con Mikrotik". "Mikrotik". [en línea], 2007, United States of America, pp. 5 – 11. [Consulta: 12 febrero 2021]. Disponible en: https://mum.mikrotik.com//presentations/AR07/mum-arg-diffserv-maxi-1.pdf.

Escalante, Mauro. *Control de Tráfico, Firewall y QoS con MikroTik RouterOS*. Lugar, España: Academy Experts, 2016, pp. 56 – 70.

IBM. "Ibm api types". "IBM". [en línea], 2007, United States of America, pp. 19 – 41. [Consulta: 23 de febrero 2021]. Disponible en: https://developer.ibm.com/apiconnect/ documentation/api\discretionary {-}{}{}101/types-apis/

Maddox, S. "Feathers api types". "Wordpress". [en línea], 2016, United States of America, pp. 20 – 32. [Consulta: 24 marzo de 2021]. Disponible en https://ffeathers. wordpress.com/ 2014/02/16/api\discretionary{-}{}{}types/

Merola. Antonio. "El ICMP, uso y abuso". "Organización Hakin9", [en línea], 2006, Italia, pp. 44 – 51. [Consulta: 24 febrero 2021]. Disponible en: http://amerola.altervista.org/doc/icmp.pdf

PHP Group. "PHP: Choosing an API – Manual". "PHP group". [en línea], 2016, United States of America, [Consulta: 25 marzo 2021]. Disponible en: http://php.net/ manual/en/mysqlinfo.api.choosing.php

Plaza, S. E, Ramírez, N. L, Acosta, C. M. API de servicios web orientados a accesibilidad. Universidad Complutense de Madrid, España. 2016. pp. 48 – 61.

Robayo. J, Darío. J, Paredes. C & Bolívar. X. Implementación de una Hotspot con servidor Radius en la Biblioteca de la Ciudad y la Provincia, Ubicada en Ambato Tungurahua (Tesis de pregrado). Pontificia Universidad Católica del Ecuador, Ambato. 2010. pp. 40 – 56.

Zamora, Andrea C. Evaluación para el Control de Tráfico y QoS en el entorno de redes de datos mediante Tecnología MIKROTIK. (Tesis de pregrado). Universidad católica de Santiago de Guayaquil, Guayaquil. 2014. pp. 40 – 46.

Ravoof. S. "Los Benchmarks Definitivos de PHP 5.6, 7.0, 7.1, 7.2, 7.3, 7.4 y 8.0". [en línea], 2021, United States of America, [Consulta: 25 Abril 2021]. Disponible en: https://kinsta.com/es/blog/puntos-de-referencia-php/

Athanasias, D. "Python Vs Php". [en línea], 2014, United States of America, [Consulta: 26 Abril 2021]. Disponible en: https://wiki.python.org/moin/PythonVsPhp

Anexos de configuraciones

Anexo A

Conexión de API – Router

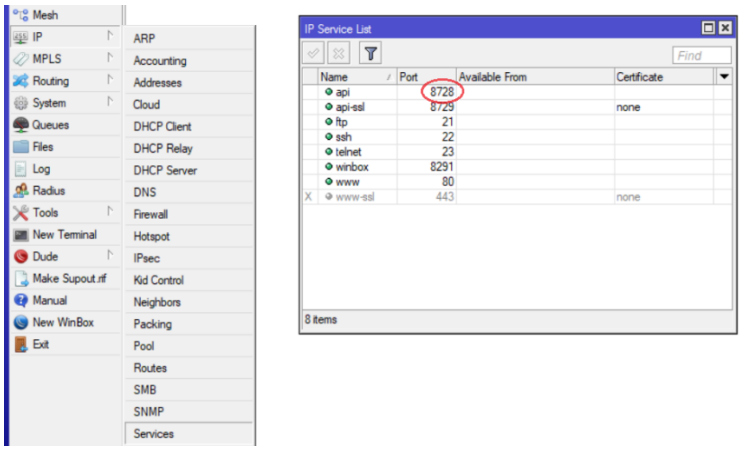
Una vez conocido lo esencial a cerca del lenguaje de programación con el que se va a trabajar que en el caso es PHP, se procedió con el proceso para la implementación de la interfaz de programación, el primer paso es el de agregar la clase necesaria para el manejo de la API en PHP.

Como se puede observar en figura previa. El código empieza de la manera en cómo se estudió previamente el inicio de un documento PHP, y seguido de la declaración de la clase para la implementación y manejo de la API de Mikrotik. Sin esta declaración no se podría tener el control del router con el que se va a trabajar.

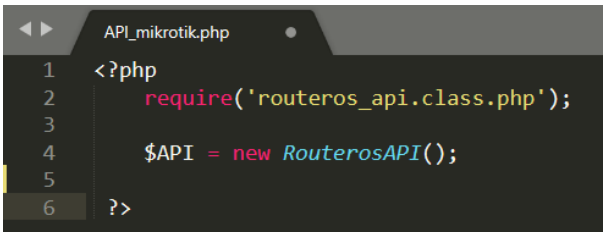
Esta clase es la provista por Mikrotik para el manejo de la interfaz de programación, y la misma que sirve para la conexión con el router. Por lo que dentro de la clase está especificado el puerto por el cual se va a conectar router.

```
class RouterosAPI
{
    var $debug     = false; //  Show debug information
    var $connected = false; //  Connection state
    var $port      = 8728;  //  Port to connect to (default 8729 for ssl)
    var $ssl       = false; //  Connect using SSL (must enable api-ssl in I
```

El puerto por defecto es el 8728. El mismo que se podría cambiar en el caso de existir puertos bloqueados. De esta manera hay que cerciorarse que en el router Mikrotik este puerto coincida con el especificado en la clase declarada. Por lo que a través de la herramienta Winbox se ingresa al router y en el menú “IP/Services”, el servicio API deberá estar con el número de puerto 8728.

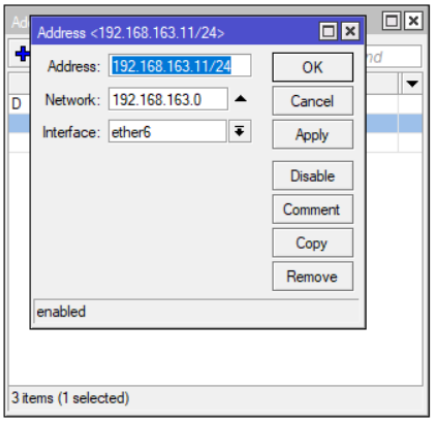


El siguiente paso dentro del documento PHP después de haber asegurado el puerto de conexión es el de crear una nueva conexión con la API de RouterOS a través de la línea de código indica a continuación.



El comando new RouterosAPI() es el encargado de la creación de una nueva conexión con el router por lo que es indispensable para poder lograr la conexión y se le asigna a una variable que se la llamara $API teniendo en cuenta la sintaxis que todas las variables se declaran con el signo "$".

A partir de esta variable $API es donde se manejará todos los comandos y configuración que serán enviadas al router. Como primera instancia son los datos del router al cual se desea acceder, los mismo que son: dirección IP del router, nombre de usuario y contraseña, en el caso de tener la configuración por defecto el equipo, el nombre de usuario será "admin" y la contraseña en blanco. Mientras que la dirección del equipo será la que se la ha asignado para la conexión con el servidor la cual se puede verificar a través de la Winbox.



Teniendo la dirección de la interface del equipo al cual se desea acceder todos los datos para acceder mediante la API están completos por lo que el comando iría de la siguiente manera mostrada en la siguiente figura:

El comando “connect” seguido de las credenciales del router es el encargado de establecer la conexión por lo que como primer campo del comando ira la dirección IP del equipo entre comillas simples, seguido del usuario y seguido de la contraseña todos separados por comas. Un control necesario que se debe hacer es el de comprobar si las credenciales del router al que se desea acceder son válidas y se logró la conexión al equipo. Por lo que se hace uso de una función “If” especificando el caso en el que se obtuvo conexión y cuando no.

De modo que cuando la condicional sea verdadera se pueda proceder al envió de comandos a través de la API cuyos resultados si se obtienen la variable que se ha denominado $data tomara el valor de 1, por otro lado, si la condicional es falsa, se establecerá que la variable $data donde se almacenara la información obtenida del router en un estado de 0.

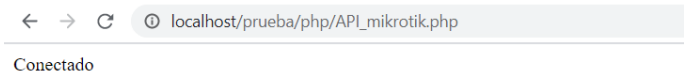
Como primera prueba de conexión al router se imprimirá en pantalla la palabra “Conectado” en el caso de que la condición del condicional If sea verdadera, siguiendo los siguientes comandos detallados a continuación:

API_mikrotik.php

```php
<?php
    require('routeros_api.class.php');

    $API = new RouterosAPI();
    $data = new StdClass();

    if ($API->connect('192.168.163.11','admin','')) {

            $ARRAY = 'Conectado';

                echo($ARRAY);
                $data->estado = 1;

    } else {
             $data->estado = 0;
    }

?>
```

De esta manera accediendo al navegador e ingresando a través del servidor XAMMP al documento PHP se puede verificar si se logró la conexión al router. Ingresando a la dirección de localización del documento PHP se podrá correr el documento.

El nombre del documento sobre el que se está trabajando es “API_mikrotik”, y solo queda especificar la dirección de ruta del mismo como se puede observar en la figura a continuación. El siguiente paso es el de ingresar a la dirección y comprobar si el mensaje de “Conectado” que se configuro en el caso de obtener conexión aparece en pantalla.



Anexo B

Datasheet de dispositivo Mikrotik utilizado

RB951Ui-2HnD

The RB951Ui-2HnD is a wireless SOHO AP with a new generation Atheros CPU and more processing power. It has five Ethernet ports, one USB 2.0 port and a high power 2.4GHz 802.11b/g/n wireless AP with antennas built in.

It has a 600MHz CPU,128MB of RAM and PoE output function for port #5 - it can power other PoE capable devices with the same voltage a s applied to the unit. Maximum load on the port is 500mA.

Package contains RouterBOARD 951Ui-2HnD in a plastic case and power adapter. Specific frequency range may be limited by country regulations.

Features	RB951Ui-2HnD (USB, power injector, USB, 2GHz, 802.11n, dual chain)
CPU	Atheros AR9344 600MHz CPU
Memory	128MB DDR2 onboard memory
Ethernet	Five independent 10/100 Ethernet ports
LEDs	Power, NAND activity, 5 Ethernet LEDs, wireless activity LED
Power in	PoE: 8-30V DC on Ether1 (Non 802.3af). Jack: 8-30V DC
Power out	PoE passive on port 5, same voltage as input.
Dimensions	113x138x29mm
Weight	Without packaging and PSU: 232g, full weight in package: 420g
Power consumption	Up to 7W
Operating Temp	-20C .. +50C
Operating System	MikroTik RouterOS, Level4 license
Package contains	RouterBOARD in a plastic case, 24V 0.8A power adapter

Antennas	2x2 MIMO PIF antennas, max gain 2.5dBi
RX sensitivity	802.11g: -96dBm @ 6Mbit/s to -80dBm @ 54Mbit/s 802.11n: -96dBm @ MCS0 to -78dBm @ MCS7
TX power	802.11g: 30dBm @ 6Mbps to 25dBm @ 54 Mbps 802.11n: 30dBm @ MCS0 to 23dBm @ MCS7
Modulations	OFDM: BPSK, QPSK, 16 QAM, 64QAM; DSSS: DBPSK, DQPSK, CCK

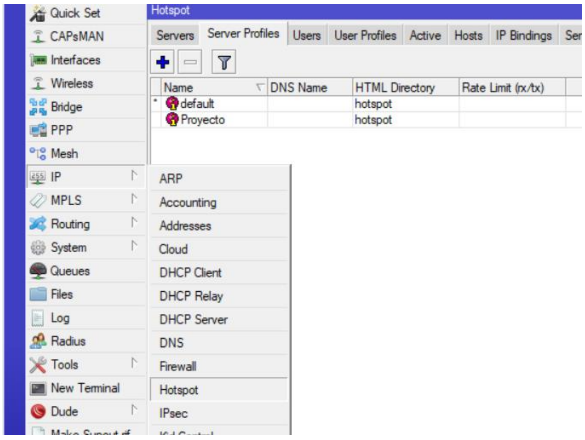
MikroTik

Anexo C

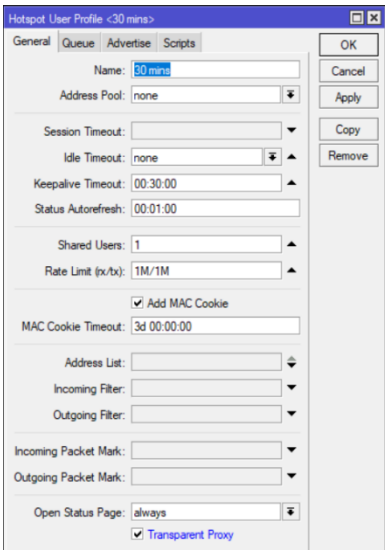
Automatización de manejo de credenciales de usuario en Hotspot

Una de las funcionalidades que ofrece la Interfaz de programación (API) de Mikrotik, es la de gestionar o administrar los usuarios del mismo. Como se pudo observar previamente, la API funciona a través del envió de líneas de código hacia el router para realizar operaciones. Por lo que para lograr el objetivo planteado se necesita conocer la sintaxis para los comandos en el manejo de usuarios del router Mikrotik.

Como se pudo apreciar previamente como primer paso está el de obtener conexión entre la interfaz de programación y el Routerboard rb951ui-2hnd. Siguiendo los pasos estudiados, y comprobando la conexión, se puede proseguir con el resto de pasos que conllevara el punto a tratar es decir llevar a cabo el proceso de administración de usuarios en Hotspot, para cual como punto de inicio será el proceso de creación propiamente del Hotspot. A través de la herramienta Winbox, se puede realizar el proceso de creación, accediendo al menú IP/Hotspot, se registra una nueva configuración de Hotspot, donde se especificará el puerto de la interface del Hotspot, aquel el cual va a tener conexión con los usuarios, es decir al cual tendrá conexión la antena de expansión de zona que funcionará como punto de acceso.



Una vez habiendo configurado el nuevo Hotspot como siguiente paso es el de registrar un nuevo perfil o perfiles, entre los cuales se van a diferenciar los usuarios, estos perfiles especificaran el tiempo de uso que tendrá el usuario, así como velocidades máximas de subida y bajada. Accediendo a la pestaña “User Profiles” se puede registrar un nuevo perfil con los parámetros que se pueden observar a continuación:



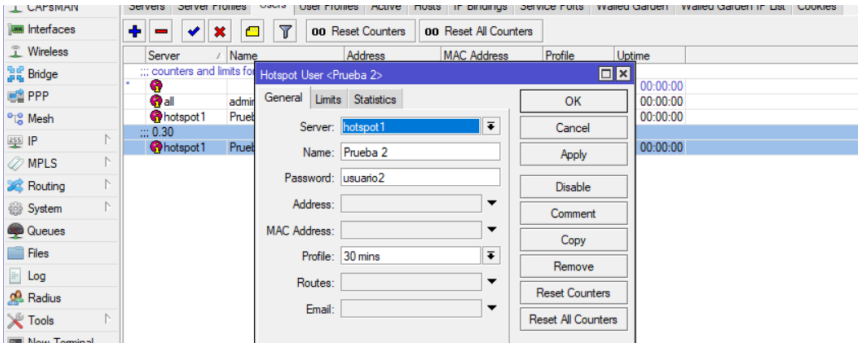
La configuración del documento PHP para poder crear usuarios de manera automatizada, comienza por la creación de un simple usuario, para poder desencadenar en un proceso más complejo de la creación automatizada. Durante la creación de usuarios a través de la herramienta Winbox se puede observar los campos necesarios, los mismo que se detallaron previamente en la Fig. 32. Por lo tanto, como primer punto del documento será la creación de variables donde se alojarán los valores que se asignarán al nuevo usuario. La sintaxis del comando "ip/hotspot/user/add" permitirá que el usuario que se va a crear se agregue al Routerboard rb951ui-2hnd. De tal manera que, siguiendo los pasos previos para el ingreso de un nuevo comando en el router, y creando las variables pertinentes, se puede obtener el siguiente documento.

API_mikrotik.php

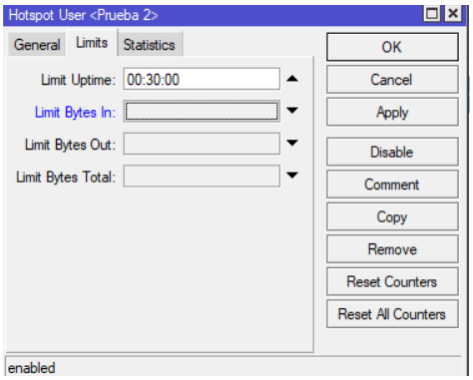
```php
<?php
    require('routeros_api.class.php');

    $API = new RouterosAPI();
    $data = new StdClass();
    $username = 'Prueba 2';
    $password = 'usuario2';
    $profile = '30 mins';
    $tiempo = '1800';
    $precio = '0.30';

    if ($API->connect('192.168.163.11','admin','')) {

            $API->write("/ip/hotspot/user/add",false);

                $API->write("=name=".$username,false);
                $API->write("=password=".$password,false);
                $API->write("=profile=".$profile,false);
                $API->write("=server=".'hotspot1',false);
                $API->write("=limit-uptime=".$tiempo,false)
                $API->write("=comment=".$precio,true);
```

Con respecto al parámetro "limit-uptime" el formato en el que se trabaja es en segundos por lo que se debe transformar la cantidad en minutos a segundos en el caso del ejemplo 1800 s que representaría los 30 minutos.

Ejecutando el documento ilustrado en la figura siguiente. Se puede esperar la creación del usuario que se ha propuesto y para comprobarlo se puede aplicar las formas previamente estudiadas, a través de la herramienta Winbox o a través de la interfaz de programación. Ingresando a la pestaña de “Users” en Winbox se puede comprobar que el usuario ha sido o no creado correctamente.



De manera instantánea el usuario ha sido registrado correctamente en el Routerboard rb951ui-2hnd. Y en la ilustración se puede observar que los campos esperados son los que se configuro en el documento PHP. Además, ingresando a la sub pestaña “Limits” se puede comprobar que la conversión de los 1800 s se realiza de manera automática, tomando el router como los 30 minutos esperados.



Aplicando en el documento PHP la visualización de usuarios del Hotspot para poder obtener los usuarios registrados en el Sistema quedaría de la siguiente manera como observa en la figura a continuación.

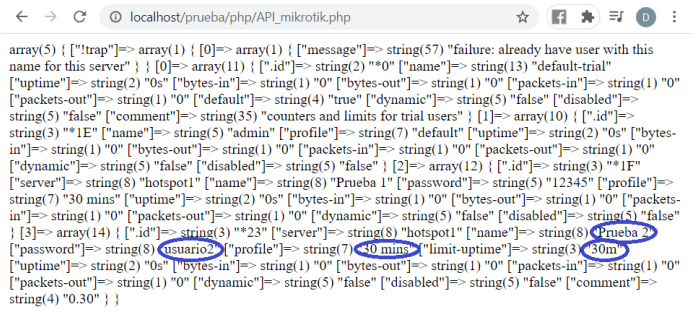
API_mikrotik.php

```php
<?php
    require('routeros_api.class.php');

    $API = new RouterosAPI();
    $data = new StdClass();
    $username = 'Prueba 2';
    $password = 'usuario2';
    $profile = '30 mins';
    $tiempo = '1800';
    $precio = '0.30';

    if ($API->connect('192.168.163.11','admin','')) {

            $API->write("/ip/hotspot/user/add",false);

                $API->write("=name=".$username,false);
                $API->write("=password=".$password,false);
                $API->write("=profile=".$profile,false);
            $API->write("=server=".'hotspot1',false);
                $API->write("=limit-uptime=".$tiempo,false);

                $API->write("=comment=".$precio,true);

        $API->write('/ip/hotspot/user/print');

        $READ = $API->read(false);
        $ARRAY = $API->parseResponse($READ);

           $API->disconnect();
           var_dump($ARRAY);

           $data->estado = 1;
```

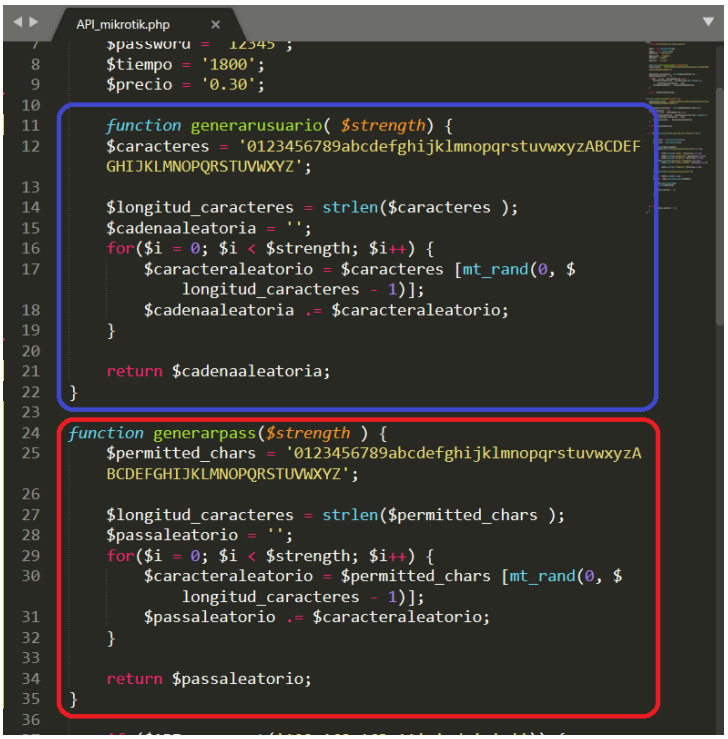
Figura 26-2: Creación y comprobación de usuario a través de API

Fuente: Leon Daryel.

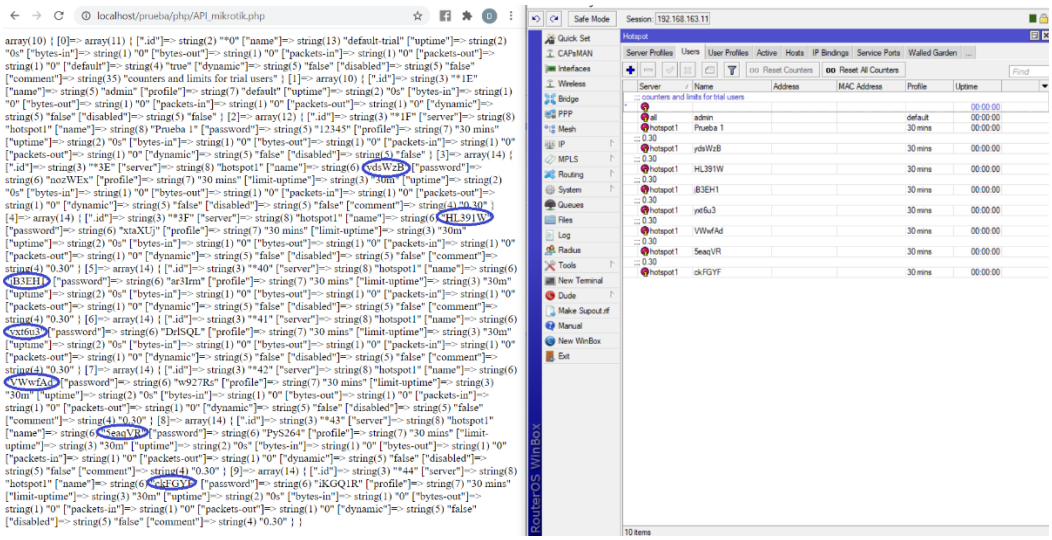
Ejecutando el documento a través del servidor en el navegador se obtiene el siguiente resultado, donde de color azul de puede observar encerrado los parámetros que se establecieron a través de código que ahora forman parte del usuario creado.



Sin embargo, a pesar que, mediante la interfaz de programación, se puede crear usuarios de manera más rápida, aun se presenta la problemática en el caso de tener que crear varios usuarios al mismo tiempo, por lo que se debe incluir funcione capaces de generar los nombres de usuarios y contraseñas de manera que se puede automatizar la creación de varios usuarios a la vez. Una de las funciones de PHP que se va a utilizar es la de “mt_rand” cuyo resultado es un numero aleatorio en un rango determinado. Esto es necesario para la creación de usuarios aleatorios, así como las contraseñas de cada uno.



Como se puede observar en la figura previa de color azul encerrado esta la función denominada “generarusuario” que es la encargada de generar un nombre de usuario aleatorio a partir de un numero de longitud que se desee. De igual manera en el recuadro rojo esta la función de creación de la contraseña aleatoria para el usuario requerido. De esta manera se podría crear usuarios sin la necesidad de especificar para cada uno un nombre de usuario y contraseña logrando así que en cada ejecución del documento PHP se genere un nuevo usuario en el Routerboard rb951ui-2hnd. Como se puede comprobar a continuación se encuentran tanto el documento PHP ejecutado 7 veces de tal manera que se generaron 7 usuarios, como la ventana de Winbox donde se encuentran dichos 7 usuarios generados de manera aleatoria. Donde en la parte izquierda se encuentra la ejecución del documento y encerrados de color azul los nombres de usuarios generados aleatoriamente que se pueden comprobar de la parte derecha con la ventana de Winbox.



Por ultimo queda la agregación de la funcionalidad de generación de varios usuarios al ejecutar una sola vez el documento PHP, a pesar que es solo cuestión de agregar otra estructura de control al documento PHP, esta ahorraría aún más tiempo haciendo el proceso más eficaz. Como primero se deberá agregar una nueva variable que se la denominará para el ejemplo “$numerous” el cual albergara el número de usuarios que se desea crear al ejecutar una sola ves el documento PHP es decir la interfaz de programación API.

```
43
44      for($i = 0; $i < $numerous; $i++) {
45
46          $username = generarusuario(6);
47          $password = generarpass(6);
48
49          if(!empty($username)){
50              $API->write("/ip/hotspot/user/add",false);
51
52                  $API->write("=name=".$username,false);
53                  $API->write("=password=".$password,false)
                        ;
54                  $API->write("=profile=".$profile,false);
55              $API->write("=server=".'hotspot1',false);
56                  $API->write("=limit-uptime=".$tiempo,
                        false);
57                  $API->write("=comment=".$precio,true);
58
59          }
60      }
61
```

Con la agregación de esta variable solo resta especificar el valor de usuarios que se desees crear a dicha variable, en el caso del ejemplo se ha especificado el valor de 5, es decir que, de una sola ejecución, la API debería generar 5 usuarios con nombres y contraseñas de manera aleatoria.

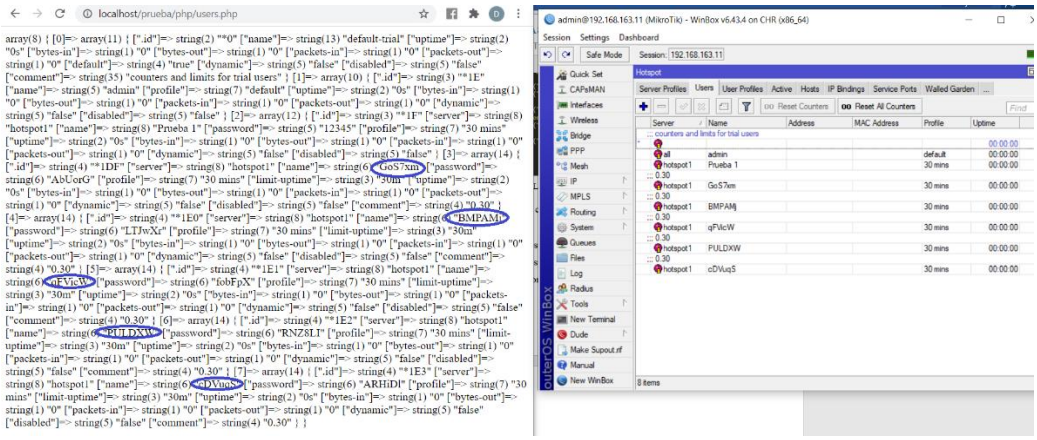


Figura 31-2: Comprobación de creación de usuarios mediante interfaz de programación automatizada

Fuente: Leon Daryel.

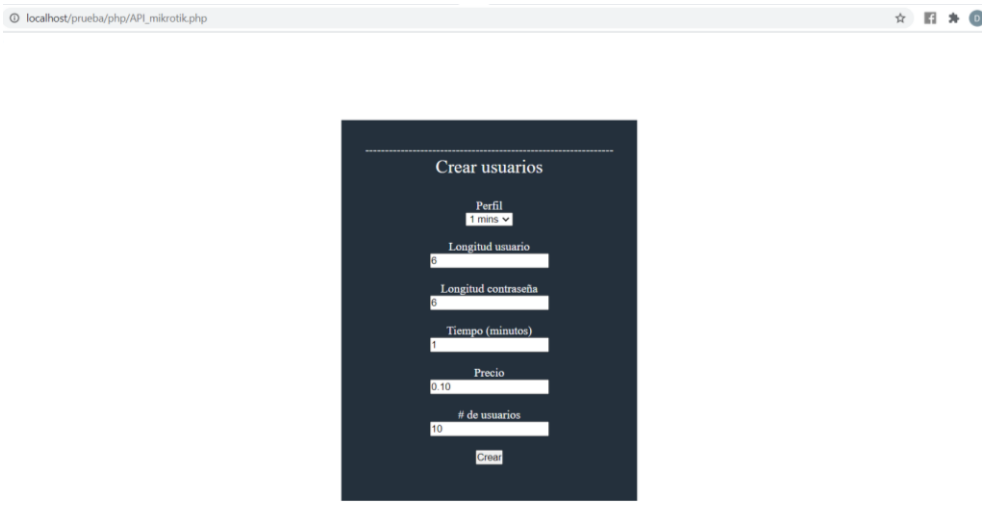
Observando la figura previa, se puede comprobar que al ejecutar una sola vez el documento PHP es decir la API, se generaron los 5 usuarios que se especificó como ejemplo, además que mediante la API y la herramienta Winbox se pueden visualizar dichos usuarios.

De esta manera solo quedaría modificar los parámetros de la interfaz de programación con respecto al usuario que se desee crear, es decir, cambiar la longitud de los nombres de usuarios o contraseñas, cambiar el perfil al cual se desea adjuntar un usuario, cambiar el servidor Hotspot, modificar el límite de tiempo de un usuario en particular o a su vez modificar el número de tickets que se desea crear en una sola ejecución. Todos estos parámetros que se pueden modificar forman parte del código PHP de la API, a manera de variables. Mediante la incursión de plantillas se puede hacer además que la API adquiera una interfaz visual, de manera que sea más amigable con el usuario administrador a la hora de administrar sus usuarios.

Incluyendo en archivo PHP el código necesario HTML de manera que se pida al usuario administrador los datos de los clientes que desee crear. En la figura a continuación. Se puede observar los campos donde el usuario ingresara los datos. Cada espacio solicitado sirve para especificar al usuario que se desea crear, por ejemplo, longitud de nombre de usuario, longitud de contraseña que otorgara al usuario, tiempo, precio y, por último, el número de usuarios que desea crear de un solo clic. De esta manera al administrador le bastaría con completar los datos solicitados para crear credenciales de usuarios en el router.

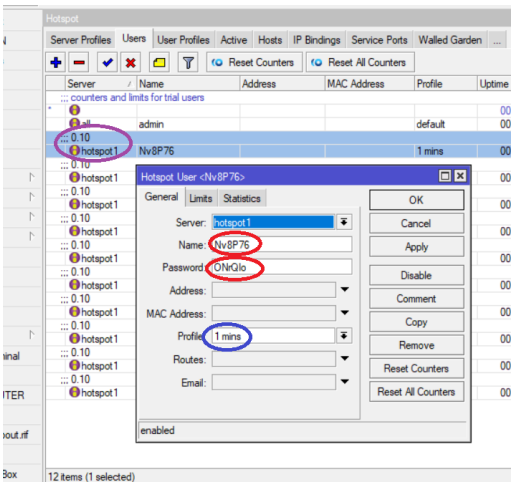
```
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="UTF-8">
    <title>Creacion de usuarios </title>
</head>
<body>
    <p>----------</p>
    <p>Crear usuarios</p>
    <form id="form1" name="form1" method="post" action="">
    <p>Perfil <br />
    <select type="text" name="numero1" id="numero1">
        <option value="1 mins">1 mins</option>
        <option value="60 mins">60 mins</option>
    </select>
    <br />
    Longitud usuario <br/>
    <label for="numero2"></label>
    <input type= "text" name="numero2" id="numero2"  value="<?=$lusername?>" />
    <br />
    Longitud contraseña <br/>
    <label for="numero3"></label>
    <input type= "text" name="numero3" id="numero3" value="<?=$lpassword?>" />
    <br />
    Tiempo (minutos) <br/>
    <label for="numero4"></label>
    <input type= "text" name="numero4" id="numero4" value="<?=$tiempo?>" />
    <br />
    Precio <br/>
    <label for="numero5"></label>
    <input type= "text" name="numero5" id="numero5" value="<?=$precio?>" />
    <br />
    # de usuarios <br/>
    <label for="numero6"></label>
    <input type= "text" name="numero6" id="numero6" value="<?=$numerous?>" />
    <br />
    </p>
```

Poniendo en marcha el archivo PHP, se obtiene la siguiente pantalla:



Donde en “Perfil” se podría especificar todos los perfiles que se pre ha pre configurados en el router, en este caso el perfil pre configurado es el de “1 min”, y completando los demás campos.

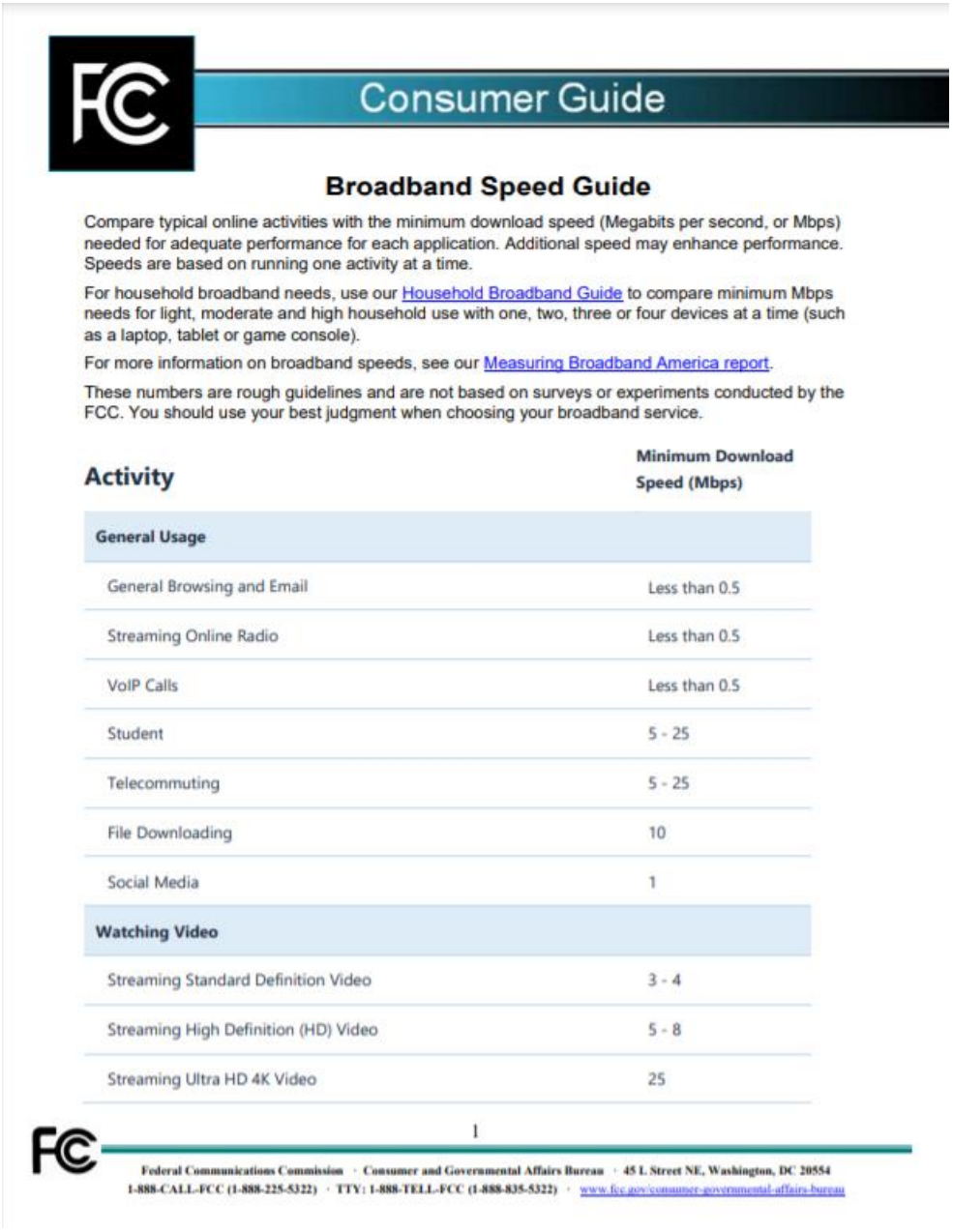
Donde procediendo a crear los usuarios mediante el botón “crear”, se puede observar en el router Mikrotik para comprobar si los usuarios fueron creados, y si contienen la información especificada durante su creación.



A partir de la figura previa, se puede observar cómo los 10 usuarios especificados durante la creación se crearon correctamente. De color morado se puede observar el precio especificado y el servidor Hotspot. De color rojo la longitud que debe tener el nombre de usuario y contraseña con caracteres aleatorios. Y por último de color azul se puede observar el perfil especificado. Como extra se puede realizar una página más amigable con el usuario incluyendo información o colores del negocio donde se desee implementar el Hotspot.

Anexo D

Guía de velocidad de ancho de banda según FCC (Federal Communications Commission)



Consumer Guide

Broadband Speed Guide

Compare typical online activities with the minimum download speed (Megabits per second, or Mbps) needed for adequate performance for each application. Additional speed may enhance performance. Speeds are based on running one activity at a time.

For household broadband needs, use our Household Broadband Guide to compare minimum Mbps needs for light, moderate and high household use with one, two, three or four devices at a time (such as a laptop, tablet or game console).

For more information on broadband speeds, see our Measuring Broadband America report.

These numbers are rough guidelines and are not based on surveys or experiments conducted by the FCC. You should use your best judgment when choosing your broadband service.

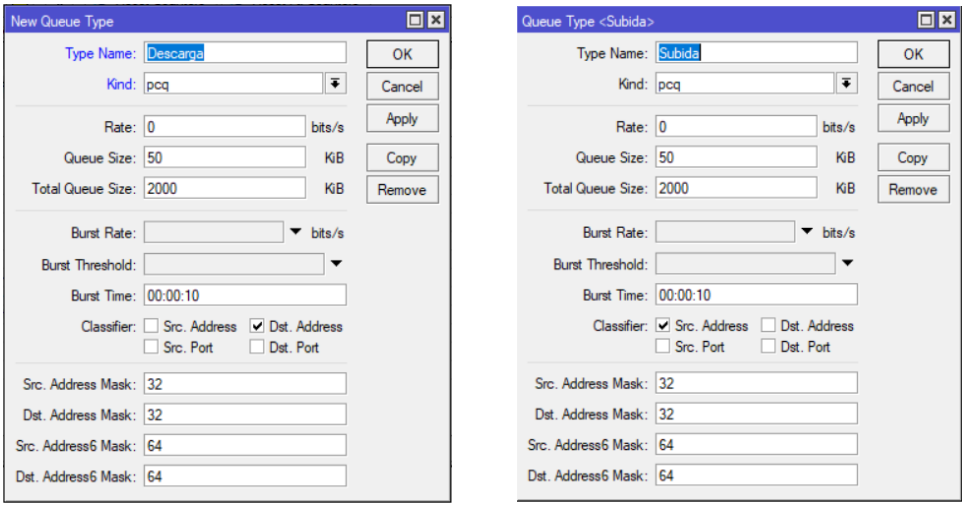
Activity	Minimum Download Speed (Mbps)
General Usage	
General Browsing and Email	Less than 0.5
Streaming Online Radio	Less than 0.5
VoIP Calls	Less than 0.5
Student	5 - 25
Telecommuting	5 - 25
File Downloading	10
Social Media	1
Watching Video	
Streaming Standard Definition Video	3 - 4
Streaming High Definition (HD) Video	5 - 8
Streaming Ultra HD 4K Video	25

Federal Communications Commission · Consumer and Governmental Affairs Bureau · 45 L Street NE, Washington, DC 20554
1-888-CALL-FCC (1-888-225-5322) · TTY: 1-888-TELL-FCC (1-888-835-5322) · www.fcc.gov/consumer-governmental-affairs-bureau

Anexo E

Implementación de control de ancho de banda

Como primer punto está la creación de las colas padres de las cuales derivaran el resto de las colas hijas, para esto se debe detallar tanto de subida como de bajada. De tal manera se accedió mediante la herramienta Winbox al menú “Queues” del dispositivo Mikrotik y se agrega las colas.
Como se puede ver en la figura a continuación, las colas padres se crean con los nombres de Subida y de Descarga, teniendo en cuenta que para Subida el “Classifier” será la dirección de origen es decir la casilla “Src.Address”, mientras que para la Descarga será en base a la dirección de destino es decir “Dst.Address”



Una vez creadas las colas padres mostradas en la figura previa, se debe crear las colas hijas que se derivan de las colas padres del árbol de Queues. En dichas colas hijas, se debe especificar la tasa de ancho de banda que se asignara a cada usuario de los perfiles, ya que sin especificar, un solo usuario podría acaparar todo el ancho de banda disponible consiguiendo ralentizar a los demás usuarios la calidad de servicio, por lo que se ha establecido un máximo de ancho de banda el cual un usuarios podría consumir, este valor dependería del valor máximo provista por el proveedor del servicio, por lo que el administrador del Hotspot seria quien especifique dicho valor, para el trabajo propuesto se especificara un máximo de 2 Mbps que podrá ser consumido entre todos los usuarios, y dotando a cada usuario de un máximo de 512 Kbps, cabe recalcar que estos valores dependerían del administrador y del servicio provisto para el Hotspot, donde se podría otorgar un mayor ancho de banda a cada usuarios para una mejor calidad de servicio.

Como se puede observar en la figura a continuación. se crean las colas hijas que serán representaran el ancho de manda máximo que cada usuario podrá hacer uso, por lo que en el campo “Rate” se especifica los 512k que se estableció para el presente trabajo. Como se mencionó, este representa todo el ancho de banda que el usuario podrá hacer uso, por lo que más adelante se creará más colas hijas especificando el ancho de banda máximo según el servicio. Así como en las colas padres para subida y bajada se deberá especificar el “Classifier”.

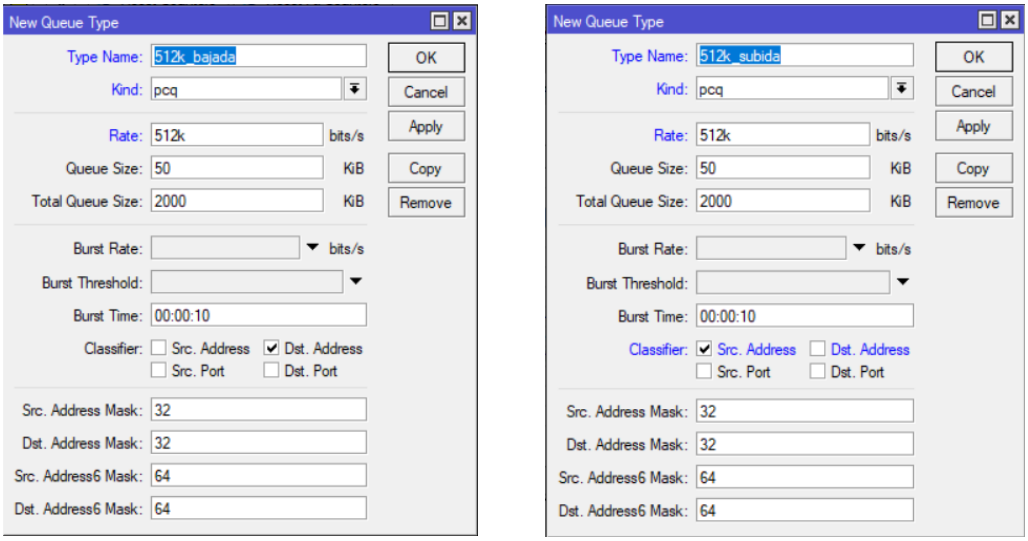
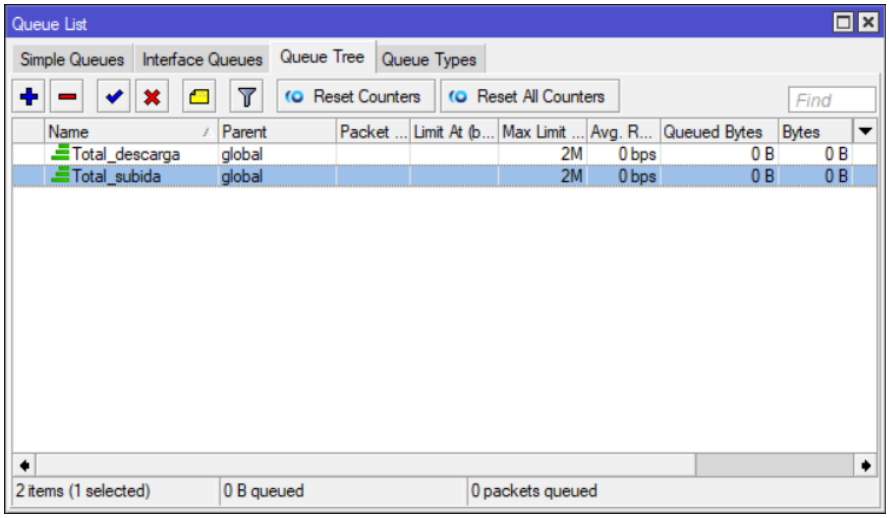


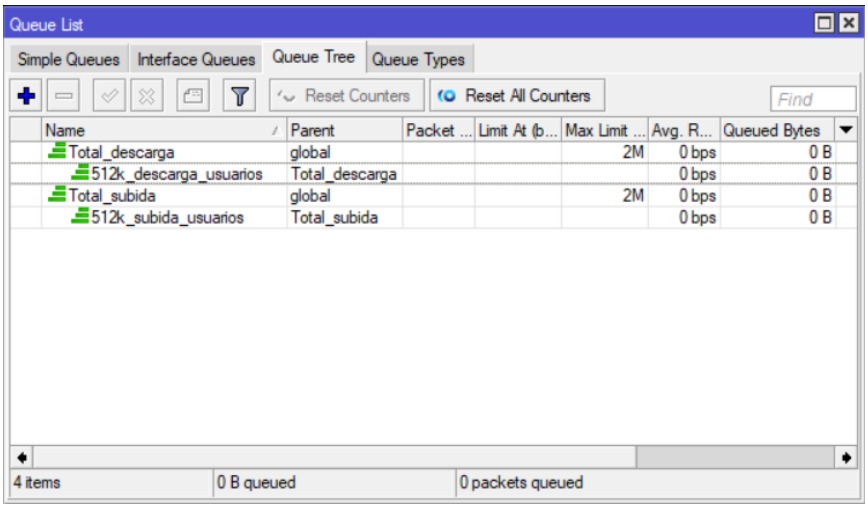
Figura 38-2: Creación de colas hijas para máximo ancho de banda por usuarios

Fuente: Leon Daryel.

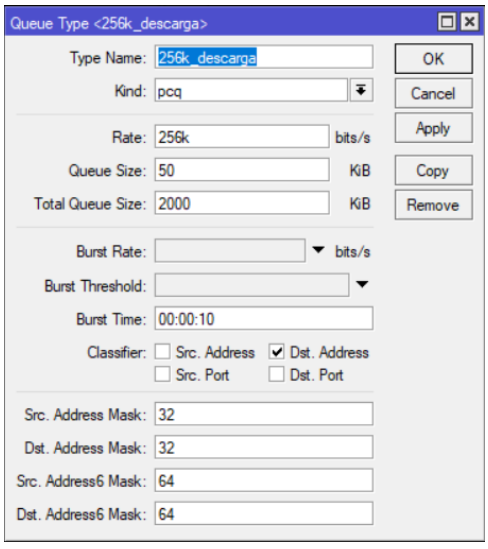
Lo siguiente fue la creación del árbol o “Queue tree” para esto se accede a dicha pestaña y se comienza la creación del árbol, especificando primero las colas padres antes creadas, y teniendo en cuenta el valor que se establecerá como máximo de ancho de banda. Como se mencionó antes para objeto del estudio es establece 2 Mbps de subida y de bajada es decir simétrico.

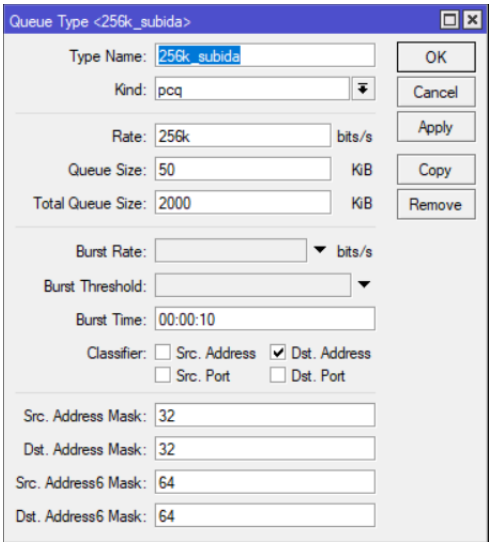


Una vez creada la mayor jerarquía del árbol, se procedió a la creación de las ramas del árbol es decir especificar el ancho de banda que los usuarios podrán disponer, por lo que en este caso el campo “Parent” será los anteriores creados de subida y bajado de esta manera.

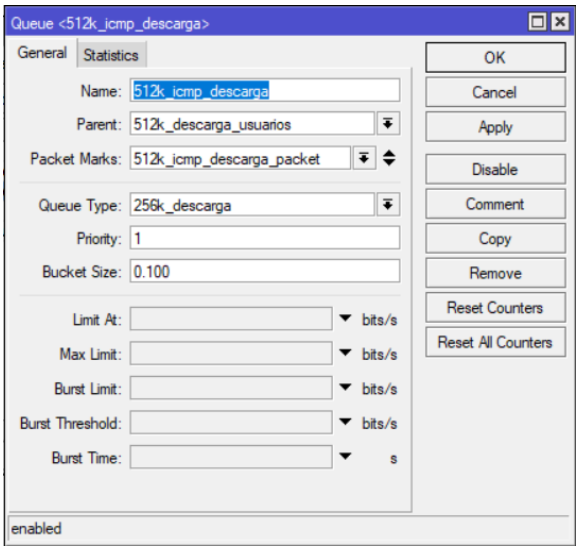


Hasta este punto se podría crear a las colas creadas la marcación de paquetes y obteniendo el control del ancho de banda entre los usuarios, sin embargo, como objetivo de estudio está el implementar políticas de calidad de servicio, por lo que se alcanza un nivel más profundo priorizando el trafico según su tipo, creando colas hijas a partir de las creadas anteriormente para cada tipo de servicio. Para esto se realiza la creación en primer lugar de un nuevo “Queue type” especificando ahora el ancho de banda máximo para los servicios como ICMP y DNS. Para estos servicios se otorga una cantidad proporcional en base al ancho de banda máximo que son los 512K, en este caso se estableció 256K al ser servicios que no necesitan un ancho de banda elevado.

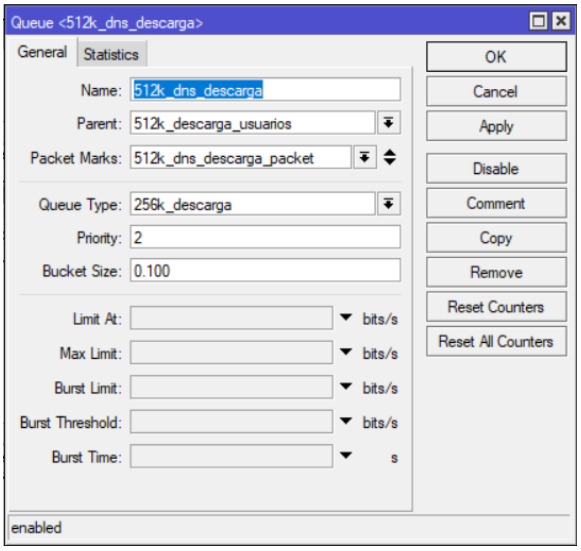




Una vez especificado el tipo de colas y el ancho de banda, se procedió a la creación de las colas hijas para la marcación de paquetes, en este punto se especifica el tipo de tráfico y teniendo en cuenta las prioridades que tendrá cada tipo de servicio, como primer punto se especificó la cola para el tráfico ICMP cuya prioridad se especificó anteriormente que sería de 1 y el "Queue type" de 256K, además que se agrega la marca que tendrá los paquetes de este tipo.

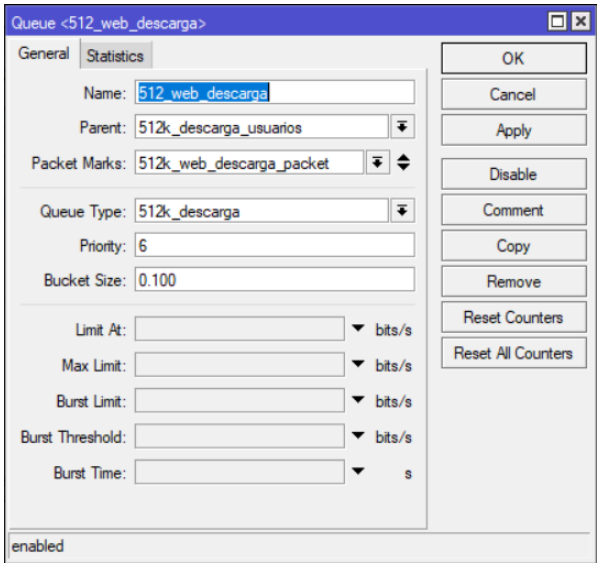


De igual manera para el tráfico DNS se especificó la cola, pero teniendo en cuenta que la marca del mismo será diferente y que la prioridad en este caso será de 2.

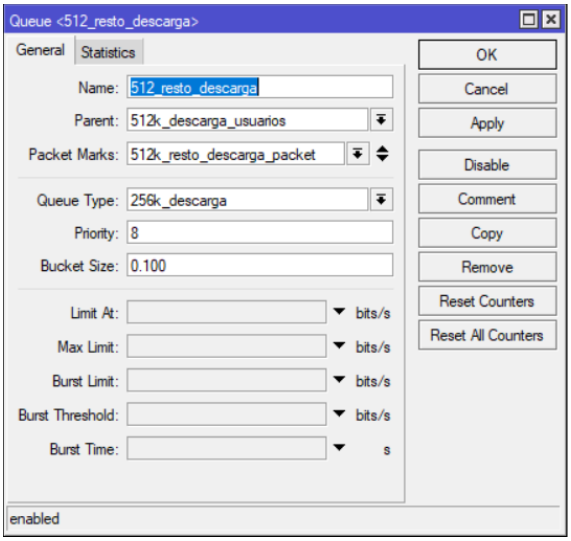


Como siguiente punto está el de especificar para el tráfico WEB que generara el usuario, por lo que en este caso se especifica todo el ancho de banda disponible, pero en este caso al ser tipo

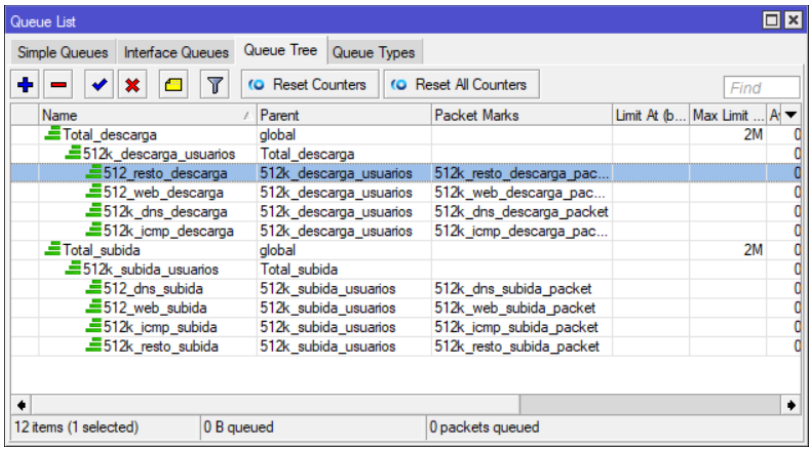
trafico HTTP, la prioridad cambia y se disminuye en este caso se colocó prioridad de 6 recalcando que se especificara otra cola con menor número de prioridad.



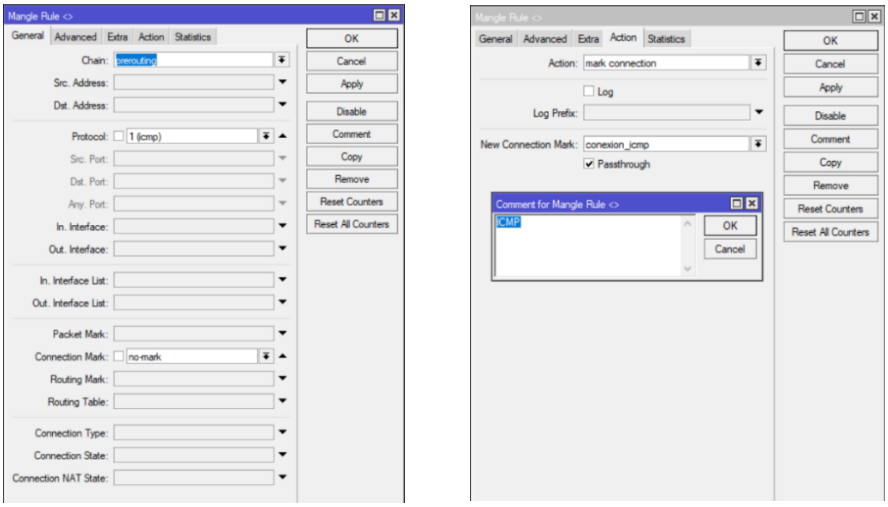
Por último, la cola que se crea es para todo el resto de tráfico que se podría generar a parte del especificado, y al ser todo el resto de trafico la prioridad es la menor, por lo tanto, se especificó 8 de prioridad y otorgando un ancho de banda proporcional al total en este caso de 256K.



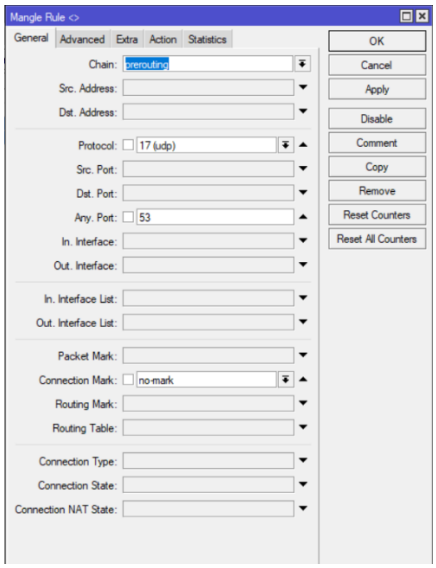
Con esto se tiene armado el árbol de bajada, ahora queda armar el árbol de subida de igual manera con los parámetros que se especificó en descarga, simplemente cambiando el “Parent” a la cola de subida, el “Queue type” y cambiado la marca que tendrán los paquetes; la figura a continuación, muestra el árbol de subida armado.

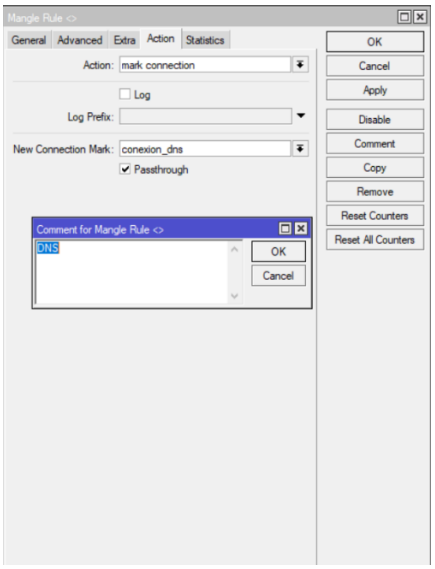


Sin embargo, para el control del tráfico se deben aplicar las marcas de paquetes para que tenga efecto el control, a través del “Mangle” del dispositivo Mikrotik, se realiza el marcado de conexión para cada tipo de tráfico especificado; de esta manera se realizó la marcación de conexión para tráfico, ICMP, DNS, WEB, y para el resto de tráfico, durante la creación de la marcación en el “Mangle” se debe tener en cuenta el campo “Chain” el cual toma el valor de “prerouting”, en el protocolo dependerá de cada servicio y según el caso se especificaría el puerto.

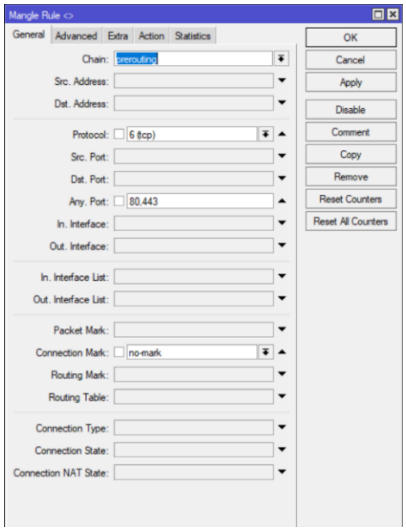


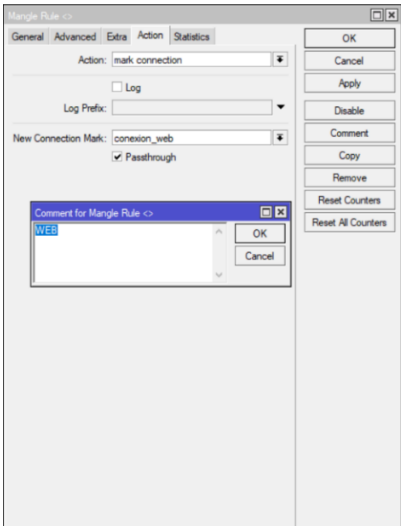
Como se puede observar en la figura previa. La regla en Mangle se especifica los campos mencionados y en este caso en particular no se especifica el puerto al ser ICMP. Mientras que para el tráfico generado DNS, además de especificar el protocolo UDP se especifica el puerto 53 que corresponde a DNS.



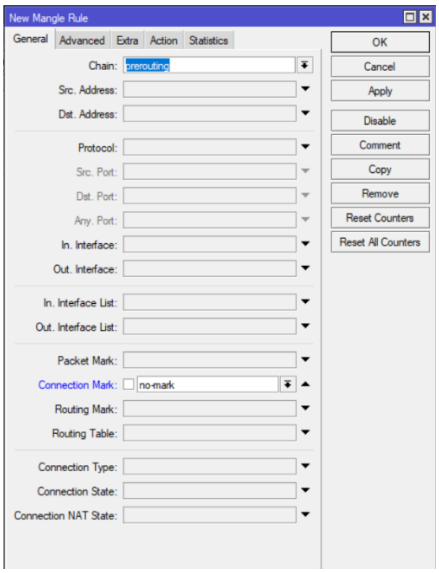


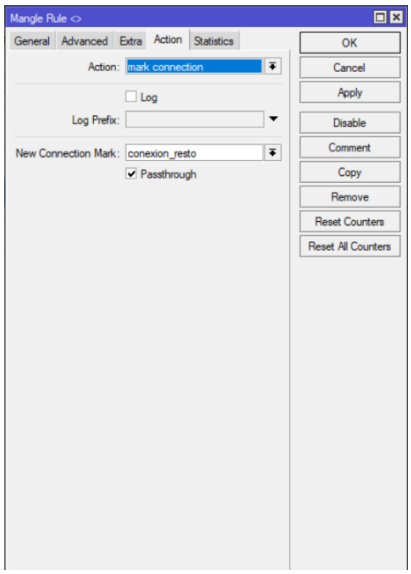
Para la conexión WEB en este caso el protocolo que utiliza es TCP y el puerto para HTTP es el número 80, mientras que para HTTPS hace uso del puerto 443, por lo que se debe especificar ambos puertos.





Para todo el resto de tráfico que se podría generar, no se especifica protocolo ni puerto y se especifica la marca que tendrán la conexión.

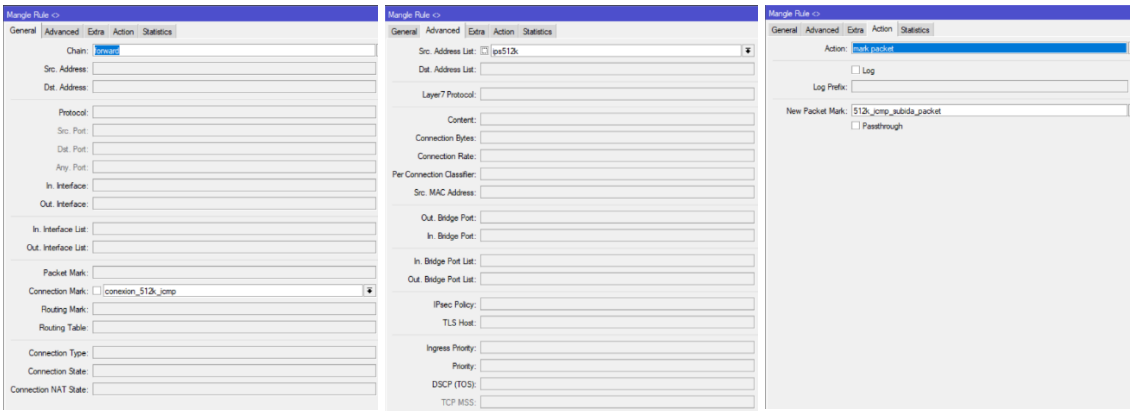




Una vez marcada las conexiones, hace falta crear en si el marcado de paquetes dependiendo el servicio; consiste en un grupo de 5 reglas en “Mangle” comenzando con la creación de la marca cuando tenga conexión al servicio, de aquí en el parámetro “Chain” especificando “Forward”, en el campo “Protocol” se deberá seleccionar el que se desee especificar, ya sea ICMP, DNS, WEB o el resto, en la marca de conexión de igual manera dependerá el servicio que se esté especificando y de las marcas de conexión antes creadas, además de seleccionar en estado de conexión como “NEW”, siguiendo hacia la pestaña “Advanced” se deberá seleccionar la direcciones de origen a aquellas especificadas en el perfil del Hotspot creado, para el presente trabajo el nombre de “ips512k”; por ultimo en la pestaña “Action” se marca la conexión con un nuevo nombre haciendo alusión al nombre de las direcciones con las que se especificó en el perfil del Hotspot.

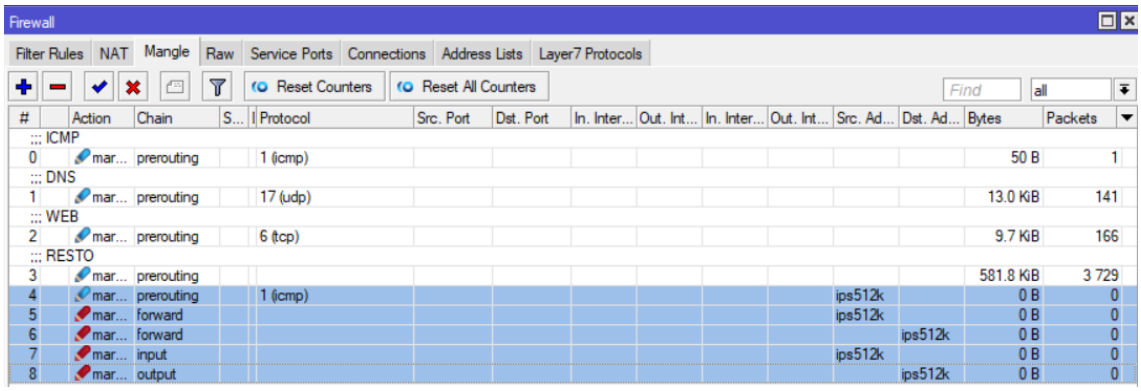
Una vez creada esta regla a partir de la cual se desarrollan las siguientes 2, esta vez para el parámetro “Chain” especificando “Forward”, y especificando una regla para las direcciones de origen del Hotspot y otra para las direcciones de destino, además que en cada regla deberá especificarse la marca de paquetes creadas en los “Queues” de tal manera que, para las direcciones de origen es especifica los de subida y para las direcciones de destino se especifica la marca de descarga.

Como se puede observar en la figura a continuación, se muestra la regla en “Mangle” como debería quedar en “Chain” con el parámetro “Forward” especificando direcciones de origen.

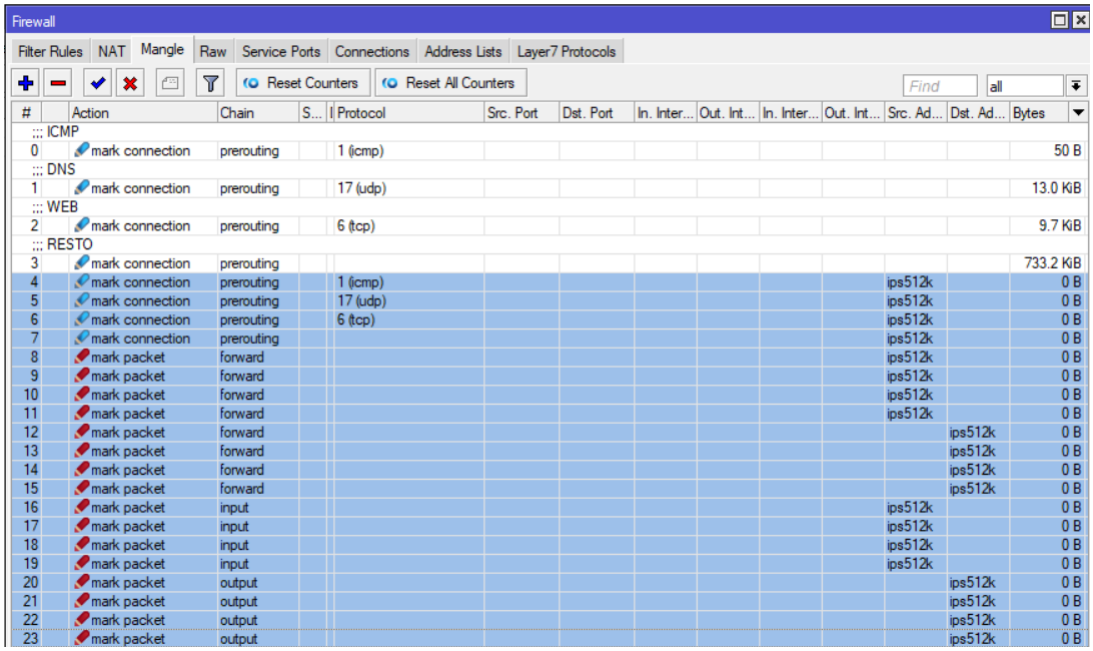


El siguiente conjunto de 2 reglas en este caso toman el valor de “Input” y de “Output” de igual manera que en “Forward” especificando para una, direcciones de origen y otra, direcciones de destino, además, de marcas de subida para una y de bajada para otra respectivamente.

La siguiente figura, muestra el resultado del conjunto de 5 reglas creadas para ICMP teniendo en cuenta las direcciones con las que el Hotspot va a trabajar. Estas 5 reglas son necesarias para la marcación de paquetes de mencionado protocolo. Por lo que, para los demás servicios o tipo de tráfico se debería hacer el conjunto de reglas de tal manera como en ICMP.



Para las reglas siguientes se deberá especificar el número de puerto ya que para ICMP no se lo realiza, y teniendo en cuenta las 5 reglas previamente creadas, otro punto importante es la especificación de la marca de paquete según el tipo de tráfico, que va en base a los “Queues” creados. Realizadas las reglas necesarias el “Mangle” final quedó de la manera como se muestra en la figura siguiente.



Anexo F

Ejemplo de toma de pruebas mediante Test de velocidad