ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"DISEÑO PARA MEDICIÓN DE EFICIENCIA DE LAS TÉCNICAS DE CALIDAD DE SERVICIO APLICADAS EN LA TRANSFERENCIA DE DATOS ENTRE REDES, MEDIANTE HERRAMIENTAS DE SOFTWARE LIBRE APLICADAS COMO ENRUTADOR DE BORDE"

INFORME DE MATERIA INTEGRADORA

Previo a la obtención del Título de:

LICENCIADO EN REDES Y SISTEMAS OPERATIVOS

DANIEL ISAÍAS ARIAS GUERRERO

PABLO ENRIQUE LÓPEZ SILVA

GUAYAQUIL – ECUADOR

AÑO: 2017

AGRADECIMIENTOS

Agradeciendo a Dios, en todo tiempo por su ayuda y por abrir las puertas para poder culminar esta etapa.

A mi esposa Ammi Berenice Astudillo de Arias y a mis progenitores Segundo Arias García y Luz Guerrero León, por haberme apoyado al largo de toda la carrera mostrándome que con perseverancia y valentía se pueden lograr grandes cosas.

Daniel Isaías Arias Guerrero

Agradezco a Dios por permitir finalizar una etapa de mi vida.

A mi madre Mariana Silva Tovar, por haberme guiado en el camino de la perseverancia, a fin de alcanzar las metas propuestas a lo largo de mi vida, por ser ella quien me han inculcado que con esfuerzo y sacrificio se puede alcanzar el éxito.

Pablo López Silva

DEDICATORIA

El presente proyecto lo dedico con gran amor a mi esposa, Ammi Berenice Astudillo de Arias, por haberme apoyado en todo momento siendo mi principal fuente de motivación para poder alcanzar las metas propuestas en mi vida.

A los profesores de la carrera por ser quienes impartieron el conocimiento necesario, a fin de llevar a cabo este proyecto.

Daniel Isaías Arias Guerrero

Este proyecto está dedicado a mis hijos, quienes son la fuente de inspiración que motivan a lograr mis metas, a fin de ser un ejemplo positivo en sus vidas y de esta manera ir dejando huella en sus vidas.

Dedico este trabajo a los profesores que impartieron su conocimiento día a día a fin de culminar este proyecto.

Pablo López Silva

TRIBUNAL DE EVALUACIÓN

.....

MSig. Robert Andrade Troya

PROFESOR EVALUADOR

.....

MSig. Albert Espinal Santana

PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

.....

Daniel Isaías Arias Guerrero

.....

Pablo Enrique López Silva

RESUMEN

En la actualidad, las comunicaciones de datos se han vuelto cada vez más esenciales para la humanidad. Contar con servicios que mantengan una alta disponibilidad, confiabilidad y tolerancia a fallos, se vuelve un gran reto en cualquier diseño de red de datos. Si se considera de manera adicional que las redes son convergentes, es decir, por un mismo medio se transmite datos, voz y video; no es una opción que el rendimiento de la red se vea afectado, por ende, se debe contar con los mecanismos suficientes que permitan una rápida recuperación. Por lo tanto, un buen sistema que permita alertar sobre cambios inusuales en la red, brindará información necesaria para la toma de decisiones.

Considerando que el área de las telecomunicaciones y sistemas informáticos es muy amplio, es preciso especificar que la solución está dirigida a controlar el tráfico de una red corporativa en el enrutador de borde. De manera que, es en este punto crítico, en donde se aplicarán los parámetros necesarios para diferenciar el tráfico de la red interna permitiendo garantizar la capacidad necesaria que los diferentes servicios requieran para llegar a su destino por medio de la Internet.

Por lo tanto, el presente proyecto pretende brindar una solución con herramientas de software libre aplicadas en un enrutador de borde, a fin de evaluar los métodos de calidad de servicio. Con estas herramientas se define cuál es la disciplina adecuada para clasificar los datos de interés. Además, disponen de funcionalidades para establecer un mecanismo de marcación del tráfico y que este se relacione con una disciplina de cola, permitiendo que el tráfico de interés pueda ser despachado según la prioridad preestablecida.

Finalmente, una disciplina de cola con clase permite realizar configuraciones de manera granular, a fin de garantizar la entrega del tráfico de interés desde una red convergente. No obstante, no se puede dejar de lado otros tipos de datos que también deben ser atendidos y puedan ser despachados con una menor prioridad, sin que esto incremente el retardo o una variación en la entrega de la información.

ÍNDICE GENERAL

AGRADECIMIENTOS ii
DEDICATORIA iii
TRIBUNAL DE EVALUACIÓN iv
DECLARACIÓN EXPRESA v
RESUMEN vi
ÍNDICE GENERAL 1
CAPÍTULO 1 3
1. DESCRIPCIÓN GENERAL. 3
1.1 Antecedentes 3
1.2 Justificación 5
1.3 Descripción del Proyecto 6
1.3.1 Problemática 6
1.3.2 Objetivo General 7
1.3.3 Objetivos Específicos 7
1.4 Alcances y Limitaciones 7
1.4.1 Alcance 7
1.4.2 Limitaciones 7
CAPÍTULO 2 8
2. METODOLOGÍA 8
2.1 Método de Investigación 8
2.2 Procedimiento y Herramientas 8
2.2.1 Procedimiento 8
2.2.2 Herramientas 9
2.3 Implementación 11
2.3.1 Requerimientos de Hardware y Recursos 11
2.3.2 Sistema operativo 12
2.3.3 Disciplina de cola sin clase 18

2.3.4 Disciplina de cola con clase.....19
2.3.5 Aplicando disciplina de cola HTB.....21
2.3.6 Creando árbol de cola HTB 21
2.3.7 Aplicando filtros 22
2.3.8 Herramientas de monitoreo 23
2.4 Plan de Implementación.....26
CAPÍTULO 3.....28
3. INTERPRETACIÓN DE RESULTADOS. 28
3.1 Resultados.....28
CONCLUSIONES Y RECOMENDACIONES. 35
BIBLIOGRAFÍA.....37
ANEXOS 38
a. GLOSARIO 38
b. CONFIGURACIÓN D-ITG 39
c. CONFIGURACIÓN SNMP.....42

CAPÍTULO 1

1. DESCRIPCIÓN GENERAL.

1.1 Antecedentes

Una red de datos permite a un grupo de computadoras compartir información y recursos entre sí. Existen varios tipos de redes y una de sus clasificaciones es por el alcance geográfico de la misma, por ejemplo, una red de área pequeña es conocida como red de área local (LAN, por sus siglas en inglés), o pueden ser redes de área extensa (WAN, por sus siglas en inglés), las cuales tienen como objetivo la transmisión de la información. Una de las redes más conocida es la Internet por su cobertura a nivel mundial.

Desde el inicio de la Internet, la transmisión de información entre redes de computadoras se manejaba con una baja tasa de transferencia. Al pasar los años, se mejoran los estándares y se crean protocolos como TCP/IP, a fin de optimizar la alta tasa de transferencia de las comunicaciones a nivel mundial y permitir la compatibilidad entre dispositivos de diferentes fabricantes [1].

Las redes de datos podrían manejar un concepto definido como “redes convergentes”. Esto pretende integrar en una misma infraestructura de comunicaciones diferentes tipos de tráfico de voz, video y datos, permitiendo abaratar los costos de ella. Utilizarla de mejor manera se vuelve un reto la necesidad de controlar este tráfico, a fin de que no exista pérdida de información, ni que la calidad de la comunicación se perciba degradada.

Como se observa en la Figura 1.1, una infraestructura de red, podría estar conformada por dispositivos de usuarios, equipos de acceso, servidores que se encargan de proveer diferentes tipos de servicios, dispositivos en la capa de núcleo que tienen la funcionalidad de enrutar el tráfico hacia un destino en particular. Y es en el enrutador de borde en el cual se enfoca este proyecto, debido a que, mediante la utilización de código abierto basado en el núcleo de Linux, se logra implementar en un computador personal las funciones de

enrutador, además de emplear técnicas de calidad de servicio para un tratamiento diferenciado de los paquetes de datos.

Figura 1.1: Dispositivos de red.

Existen aplicaciones multimedia que realizan actividades en tiempo real y no toleran un retardo elevado en la transmisión, por esta razón se utilizan protocolos que transportan los paquetes de datos y no ejecutan una retransmisión de paquetes, pudiéndose llegar a perder varios en la comunicación. En función de lo antes mencionado, para evitar que la transmisión los paquetes de datos se vean afectados por el retardo, se necesita de alguna técnica que permita garantizar y priorizar el envío y recepción de los datos de interés.

Cabe señalar, que al no utilizar algún mecanismo que clasifique el tipo de tráfico y le proporcione tratamiento diferenciado, la información que es enviada puede ser afectada, por ejemplo, en una videollamada se pueden ver degradadas las imágenes mostrando un “pixelado” y la voz puede escucharse robotizada o entrecortada. Por lo tanto, es necesario que sobre todo el tráfico de interés

generado por una aplicación se defina obligatoriamente un método de calidad de servicio adecuado [2].

Actualmente existen algunos métodos de calidad de servicio que priorizan las transmisiones de paquetes de datos, clasificándolos según el tipo de tráfico de interés, sea este voz, video o datos. Es necesario identificar el tipo de dato de interés de la empresa en donde se requiera aplicar alguna política de control, con el fin de priorizar los datos según su modelo de negocio. A manera de ejemplo, una compañía telefónica deberá brindar una óptima calidad de servicio en las llamadas telefónicas de sus abonados, por lo tanto, en este caso se identifica que el tráfico de interés de dicha empresa son los paquetes de voz. Por otra parte, una empresa de televisión tiene como prioridad la calidad de video y audio, siendo este su tráfico de interés, el cual debe ser controlado para brindar un excelente servicio.

1.2 Justificación

El desafío es garantizar que una infraestructura de red mantenga alta disponibilidad priorizando los datos de interés. Es por ello, que realizar malas configuraciones en los dispositivos que administran el tráfico, pueden perjudicar el rendimiento de la red. Por esta razón, es necesario aplicar una técnica adecuada de calidad de servicio con el objetivo de garantizar un óptimo manejo de los paquetes de datos.

Mediante la implementación de una solución basada en código abierto con el núcleo de Linux, se dispone de un abanico de posibilidades y herramientas para monitorear y controlar el tráfico de interés, asimismo, obtener estadísticas que permitan evaluar el desempeño de la red y aplicar medidas preventivas que garanticen la confiabilidad de los servicios que brinda el negocio.

Considerando las redes convergentes, se propone un diseño que identifique el tráfico de interés de una empresa, a fin de establecer prioridades y con esto permitir que se brinde un tratamiento especial y pueda mantener la calidad de servicio que se requiera.

1.3 Descripción del Proyecto

1.3.1 Problemática

Cabe destacar que con redes convergentes se encuentra todo tipo de tráfico transmitiéndose en la red y para ello existen reglas definidas de manera predeterminada que ofrecen calidad de servicio. Estas reglas se encuentran embebidas en los diferentes componentes de la red, sean estos, enrutadores, conmutadores o las respectivas aplicaciones que proveen sus servicios.

Las reglas que están configuradas de manera predeterminada no son eficientes, dado que, no evitan que se incrementen los tiempos de respuesta, ocasionando que ciertos servicios que son sensibles a tiempos de retardo elevado, pérdidas de paquetes, o incluso a una variación del tiempo en la entrega de los paquetes “jitter”, pueden afectar la calidad de los servicios ofrecidos.

Por esta razón, se considera que la latencia es un factor determinante, en la calidad de los servicios de las comunicaciones en una infraestructura de red convergente. Además, el mal uso de técnicas de priorización de tráfico, ocasiona que los datos de interés no tengan la prioridad requerida para que se garantice su entrega en los tiempos adecuados.

Si se aplica una técnica de calidad de servicio en la cual se clasifica el tipo de tráfico y se asigna un ancho de banda para los datos de interés, se debe considerar la capacidad máxima del ancho de banda que los datos de interés necesitan, para de esta manera evitar que los datos se pierdan durante la transmisión.

El equipo que va ejecutar el rol de enrutador y aplicar el método de calidad de servicio adecuado, debe cumplir con requerimientos mínimos de hardware, a fin de que no se sature mientras encola los paquetes de datos al momento de existir picos altos de tráfico en la red.

1.3.2 Objetivo General

Diseñar una herramienta de software libre que garantice la entrega del tráfico de interés hacia su destino correspondiente, manteniendo un tiempo promedio de latencia requerido para un óptimo servicio, de tal manera que aplicando las técnicas adecuadas se optimicen las comunicaciones de la empresa.

1.3.3 Objetivos Específicos

1. Operacionalizar un laboratorio que permita comparar los tiempos de retardo en la entrega de paquetes.
2. Configurar las aplicaciones y parámetros necesarios en el sistema operativo Linux que ejecutará el rol de enrutador, control de tráfico y monitoreo.
3. Sintetizar los datos obtenidos de retardos con la finalidad del establecimiento de una línea base que permita la toma de decisiones y genere indicadores.

1.4 Alcances y Limitaciones

1.4.1 Alcance

Sistematizar herramientas de software libre aplicados en un computador que tendrá el rol de enrutador de borde, a fin de analizar los diferentes tipos de retardos que existen, permitiendo emplear el mejor método de calidad de servicio para los paquetes de datos de interés de una pequeña o mediana empresa.

1.4.2 Limitaciones

Puesto que, no se dispone de un ambiente real, se trabajará en un entorno de laboratorio con simuladores de tráfico. Esto podría influir en los resultados obtenidos.

CAPÍTULO 2

2. METODOLOGÍA.

2.1 Método de Investigación

El método de investigación científica a utilizar es el empírico-experimental. La razón es que, basado en el conocimiento adquirido y apoyado en un entorno de pruebas controlado, se podrá validar la solución, y con esto crear conocimiento, basado en la experiencia por el análisis de los resultados de las pruebas realizadas.

Con el objetivo de validar los resultados esperados, en este proyecto se ha tenido que considerar los siguientes aspectos:

- Con el sistema operativo basado en el núcleo de Linux, se considera habilitar la función de reenvío de paquetes para que el equipo trabaje con el rol de enrutador.
- Configurar los diferentes servicios de la herramienta “Netfilter” para el control de tráfico y crear las reglas necesarias de marcación de paquetes.
- Levantar herramientas de monitoreo para obtener estadísticas de tráfico y sus respectivos tiempos de retardo.
- Generar el envío y recepción de paquetes de datos con D-ITG y 3CX para saturar las interfaces de red, y con esto, realizar la respectiva priorización de tráfico.

2.2 Procedimiento y Herramientas

2.2.1 Procedimiento

Para implementar el sistema de medición y control de tráfico se va a operacionalizar un laboratorio y para esto deberán seguirse los siguientes pasos:

a) Contar con los equipos necesarios para el laboratorio. Estos son: una computadora portátil que tendrá el rol de enrutador de borde, además

se contará con equipos portátiles con el rol de estaciones de trabajo. También se dispondrán de conmutadores, puntos de acceso inalámbricos y una conexión a la Internet.

b) Instalar un sistema operativo basado en el núcleo de Linux, cuya distribución a utilizar será Ubuntu server 16.04.
c) Las estaciones de trabajo tendrán sistemas operativos con núcleo Linux y otras estaciones con Windows.
d) Se crearán las redes necesarias para cada segmento de red.
e) Instalar las aplicaciones necesarias que permitan generar, monitorear y controlar el tráfico, y habilitar el servicio de telefonía.
f) Elaborar los escenarios de pruebas que permitirán validar el desempeño de cada disciplina de calidad de servicio a evaluar.
g) Analizar los resultados obtenidos para elaborar las respectivas conclusiones.

2.2.2 Herramientas

Las herramientas a utilizar serán agrupadas en:

a) Sistema operativo.
b) Monitoreo.
c) Control.
d) Servicios.

Sistema operativo. – Se utilizará la distribución Ubuntu Server 16.04, basado en el núcleo de Linux. Este sistema ofrece una gran cantidad de información y disponibilidad de herramientas adicionales, además del apoyo de la comunidad que brinda respuestas a las consultas que se realizan. También se hará uso de Windows 7 que permitirá simular la recepción de tráfico.

Monitoreo. – Se va a emplear varias herramientas necesarias para observar el comportamiento del tráfico generado en la red. A continuación, se realiza una explicación breve de cada una.

a) **IPTraf,** sirve para obtener estadísticas de red. La manera que lo hace es procesando el tráfico que pasa por la interfaz que se desea monitorear, permitiendo identificar su tipo y contar con la tasa de transferencia en tiempo real.

b) **SmokePing,** se puede monitorear el retardo de los paquetes, así como obtener la variación en el retardo y el porcentaje de pérdida.

c) **MRTG,** permite monitorear el tráfico generado en los diferentes nodos, presentando un gráfico histórico que puede ser mostrado por día, semana, mes o año.

Control. – Este grupo de herramientas permitirán etiquetar el tráfico o manipular la cabecera del protocolo Internet, para que según esto las políticas de control puedan establecer cómo priorizarán el tráfico de interés. A continuación, se describe las herramientas utilizadas:

a) **Traffic Control**, permite establecer colas para ofrecer un tratamiento especial al tráfico que se desee priorizar y con esto brindar calidad de servicio.

b) **IPTables,** es una herramienta de control de acceso muy poderosa. Entre varias de sus funciones permite establecer reglas para denegar y permitir tráfico, así como para establecer etiquetas o manipular el campo de la cabecera del protocolo de Internet.

Servicios. – existen varias herramientas que permitirán habilitar servicios en la red. Estas aplicaciones serán descritas a continuación:

a) **D-ITG (Distributed Internet Traffic Generator),** es una aplicación que genera tráfico de paquetes en una red para simular un entorno real, tiene la bondad de producir tráfico IPv4 e IPv6, las pruebas de simulación se las muestra en la sección de Anexos.

b) **3CX,** IPPBX permite establecer comunicaciones de voz sobre el protocolo Internet y cumple con las funciones de una central telefónica.

2.3 Implementación

Basado en las herramientas descritas anteriormente, se establecen las configuraciones necesarias para la implementación de un laboratorio que permita monitorear y controlar el tráfico generado por los servicios habilitados sobre el sistema operativo con núcleo de Linux en la distribución Ubuntu server 16.04.

2.3.1 Requerimientos de Hardware y Recursos

A continuación, en la Tabla 1, se detallan los requerimientos mínimos del computador que, bajo una distribución de Linux, ejecutará el rol de enrutador de borde.

Sistema Operativo	RAM	CPU	DISCO DURO	INTERFACE
Linux x64	8GB	I7	1000	100 Mbps Eth0 54 Mbps WLAN a 10 Mbps
Linux x86	4GB	I3	500	100 Mbps Eth0 100 Mbps Eth1 a 10 Mbps

Tabla 1: Requerimientos de hardware

Existen requerimientos mínimos establecidos para la voz sobre IP (VoIP). Para ofrecer un servicio de calidad de lo antes mencionado, en la Figura 2.1 se describen los factores a tener en cuenta.

Figura 2.1: Requerimientos mínimos de VoIP [3].

2.3.2 Sistema operativo

Se deberá contar con la distribución de Ubuntu Server instalado y funcionando de manera correcta (los detalles de la instalación del servidor no serán descritos en este proyecto). En la Tabla 2, se muestra la configuración de las interfaces de red debe estar de la siguiente manera, para que pueda cumplir con el rol de enrutador de borde:

Conexión	Interface	IP/Máscara	Puerta de enlace
LAN	P2P1	192.168.100.100/24	
WAN	Wlan1	192.168.1.100/24	192.168.1.1

Tabla 2: Configuración de interfaces

Desde la terminal de Linux se procederá a habilitar el reenvío de paquetes, esto permitirá que el servidor Ubuntu tenga el rol de enrutador, tal como se muestra en la siguiente línea:

- *echo "1" > /proc/sys/net/ipv4/ip_forward*

Para realizar el resto de configuraciones, es necesario tener un conocimiento previo de los parámetros que se van a establecer mediante la herramienta “IPTABLES”. Se debe aclarar que las cadenas son un conjunto de reglas contenidas en tablas, que se aplican en un momento determinado del flujo por donde circulan los paquetes [4].

Existen cinco cadenas en este entorno, las cuales son [4]:

- Prerouting: las reglas se aplican a los paquetes entrantes previo a cualquier decisión de enrutamiento.
- Input: establece las reglas sobre los paquetes entrantes que tienen como destino el sistema local.
- Forward: permite aplicar las reglas a los paquetes que ingresan por una interfaz y con destino directo a otra interfaz de red.
- Output: las reglas aplicadas en esta cadena tienen como origen el sistema local y como destino la interfaz de red saliente.

- Postrouting: permite aplicar las reglas que tengan como destino la interfaz de red. Por esta cadena pasa el tráfico originado desde el sistema local como también el reenviado desde otros nodos utilizando la cadena “Forward”.

A continuación, en la figura 2.2, se muestra el proceso de un paquete que pasa por las diferentes cadenas de “IPTABLES”.

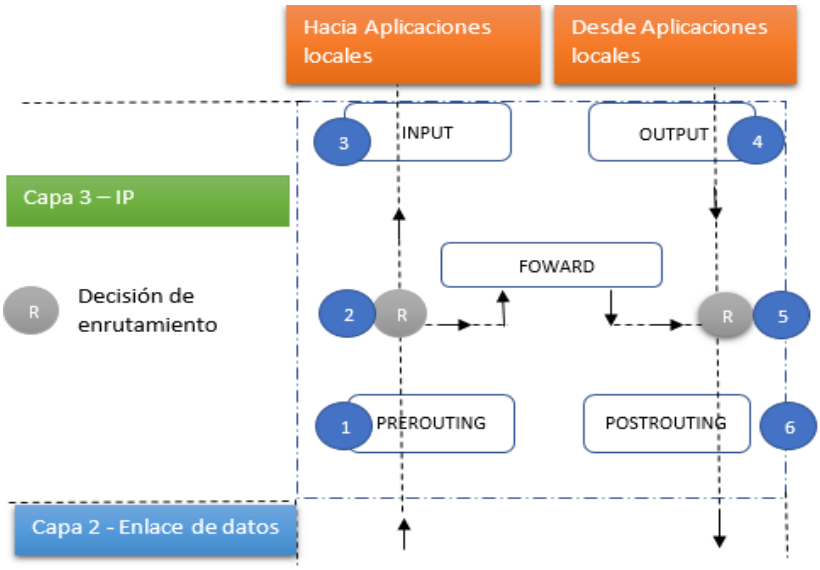


Figura 2.2: Proceso establecido en NetFilter.

Descripción del proceso [4]:

1. El paquete ingresa por la interfaz de red y la cadena “PREROUTING” aplica las reglas respectivas.
2. Si el paquete tiene como origen y destino un sistema diferente al local, las reglas que se aplican serán las definidas en la cadena “FORWARD”.
3. Si el paquete tiene como destino el sistema local, las reglas que se aplicarán serán las especificadas en la cadena “INPUT”.
4. En el caso de que el paquete se origine en el sistema local, se aplican las reglas establecidas por la cadena “OUTPUT”.

5. Si el paquete tiene como origen y destino un sistema diferente al local, las reglas que se aplican serán las definidas en la cadena “FORWARD”.
6. Para los paquetes que se encuentran previo a ser despachados por la interfaz de red cuyo origen sea el sistema local u otros sistemas se aplicarán las reglas especificadas en la cadena “POSTROUTING”.

Las tablas permiten establecer las reglas que se ejecutarán cuando sean llamadas por las cadenas. A continuación, se describe los cuatro tipos de tablas empleados en “NetFilter” [4]:

- FILTER: Utilizada para realizar un filtrado general de los paquetes, definiendo cuales ingresan o salen. Puede aplicarse en las siguientes cadenas:
  - INPUT
  - FORWARD
  - OUTPUT
- NAT: Permite modificar las direcciones de origen y destino. Las siguientes cadenas pueden aplicar esta tabla.
  - PREROUTING (DNAT)
  - INPUT (SNAT)
  - OUTPUT (DNAT)
  - POSTROUTING (SNAT)
- MANGLE: Permite modificar la cabecera del paquete con el objetivo de alterar sus campos o poder marcar etiquetas con el propósito de brindar un tratamiento diferenciado. Puede aplicarse en las siguientes cadenas:
  - PREROUTING
  - INPUT
  - FORWARD
  - OUTPUT
  - POSTROUTING

- RAW: Permite establecer excepciones al tráfico para que no sea modificado por otras reglas. Puede ser empleada en las cadenas siguientes:
    - PREROUTING
    - OUTPUT

Los parámetros a establecer en “Netfilter” van a modificar el campo “ToS” de la cabecera del protocolo Internet mediante la herramienta “IPTABLES” [5]. Este parámetro permite brindar la máxima prioridad para “DSCP” con un valor de 46 o su equivalente en hexadecimal que es “e2” [6]. A continuación, las configuraciones necesarias a realizar mediante la consola de Linux:

- *iptables -A OUTPUT -t mangle -p udp -m udp --sport 5060 -j DSCP --set-dscp 0x2e*
- *iptables -A OUTPUT -t mangle -p udp -m udp --dport 5060 -j DSCP --set-dscp* 0x2e
- *iptables -A OUTPUT -t mangle -p udp -m udp --sport 10000:20000 -j DSCP --set-dscp 0x2e*
- *iptables -A OUTPUT -t mangle -p udp -m udp --dport 10000:20000 -j DSCP --set-dscp 0x2e*

Luego se procede a marcar con una etiqueta a los paquetes de ciertos puertos donde se encuentra el tráfico de interés para que tenga un tratamiento según la prioridad que se establezca.

A continuación, se muestra el arcado del tráfico con primera prioridad para ser despachados por la cola que se establezca más adelante:

- *iptables -A FORWARD -t mangle -p udp -m udp --sport 5060 -j MARK --set-mark 100*
- *iptables -A FORWARD -t mangle -p udp -m udp --dport 5060 -j MARK --set-mark 100*
- *iptables -A FORWARD -t mangle -p udp -m udp --sport 10000:20000 -j MARK --set-mark 100*
- *iptables -A FORWARD -t mangle -p udp -m udp --dport 10000:20000 -j MARK --set-mark 100*

- *iptables -I POSTROUTING 1 -t mangle -p icmp -j MARK --set-mark 100*

A continuación, se muestra el marcado del tráfico con segunda prioridad para ser despachados por la cola que se establezca más adelante:

- *iptables -A FORWARD -t mangle -p tcp -m tcp --sport 80 -j MARK --set-mark 101*
- *iptables -A FORWARD -t mangle -p tcp -m tcp --dport 80 -j MARK --set-mark 101*
- *iptables -A FORWARD -t mangle -p tcp -m tcp --sport 443 -j MARK --set-mark 101*
- *iptables -A FORWARD -t mangle -p tcp -m tcp --dport 443 -j MARK --set-mark 101*
- *iptables -A FORWARD -t mangle -p tcp -m tcp --dport 53 -j MARK --set-mark 101*
- *iptables -A FORWARD -t mangle -p udp -m udp --dport 53 -j MARK --set-mark 101*

A continuación, se muestra marcado del tráfico con tercera prioridad para ser despachados por la cola que se establezca más adelante:

- *iptables -A FORWARD -t mangle -p tcp -m tcp --dport 25 -j MARK --set-mark 200*
- *iptables -A FORWARD -t mangle -p tcp -m tcp --dport 110 -j MARK --set-mark 200*
- *iptables -A FORWARD -t mangle -p tcp -m tcp --dport 143 -j MARK --set-mark 200*
- *iptables -A FORWARD -t mangle -p tcp -m tcp --dport 20 -j MARK --set-mark 200*
- *iptables -A FORWARD -t mangle -p tcp -m tcp --dport 21 -j MARK --set-mark 200*

En donde:

Iptables: Es el comando que permitirá modificar el campo ToS de la cabecera del protocolo de Internet.

Output: Establece la cadena de los paquetes que salen y que tienen como origen el sistema local (enrutador). Las reglas se aplican antes de tomar la decisión de ruteo.

Mangle: Es la tabla en donde se registrará los cambios establecidos en la cabecera de los paquetes.

Los demás parámetros permiten especificar el protocolo a utilizar, los puertos de origen y de destino, el valor a establecer para el campo “DSCP” especificado en hexadecimal y la marca que se establece a los paquetes que se les brindará prioridad. Cabe indicar, que los paquetes no marcados tendrán la prioridad mínima pero también serán transmitidos.

La aplicación del parámetro “46” en el campo “DSCP”, permitirá que los paquetes que transiten por los diferentes puntos intermedios tengan un tratamiento especial y sean despachados con máxima prioridad permitiendo controlar el retardo. En la Figura 2.3, se representa los diferentes retardos que existen al momento en que un paquete entra al enrutador, tales como retardo de transmisión, encolamiento y procesamiento [7].

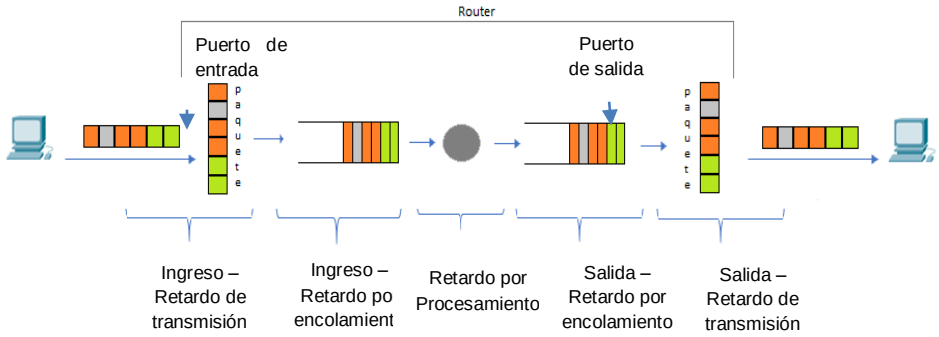


Figura 2.3: Tipos de retardos en el enrutador [7].

Luego de establecer las marcas o modificaciones de los paquetes se procede a establecer cómo serán manipulados. Con el control de tráfico se puede definir la prioridad que los paquetes tendrán y cuánto ancho de banda se les asignará, “Traffic Control” o su abreviatura “TC” utiliza cuatro métodos para manipular el tráfico, los cuales son: conformado de tráfico, reordenación de paquetes, vigilancia y eliminación de paquetes. A continuación, se muestra la actividad que hace cada uno de ellos [4]:

- Conformado de tráfico, este mecanismo es utilizado para controlar las grandes ráfagas de tráfico saliente.
- Reordenación de paquetes, sirve para que un tráfico tenga prioridad sobre otro al momento de ser transmitido.
- Vigilancia, tiene la misma funcionalidad que “Conformado de tráfico” pero afecta al tráfico de entrada.
- Eliminación de paquetes, se encarga de descartar los paquetes tanto de entrada o de salida cuando las colas están llenas.

El control de tráfico en el núcleo de Linux considera tres aspectos que forman parte del proceso de priorización de los paquetes previo al momento de transmitirlos, los cuales son: disciplina de colas, filtros y clases. Estas disciplinas de colas son conocidas en el entorno de Linux como “QDisc” (Queuing Discipline). Antes de continuar, se revisará otro tipo de colas donde no se maneja clases, pero son ampliamente utilizadas por estar habilitada de manera predeterminada en los entornos bajo Linux.

2.3.3 Disciplina de cola sin clase

Existen algunos tipos de disciplinas de encolamiento sin clase, lo que significa que no puede crearse una estructura jerárquica estableciendo una clase padre con otras clases hijas, tal es el caso de FIFO (First In First Out). En la Figura 2.4, se muestra la manera en que opera, donde el primer paquete en llegar será el primero en salir. Si el controlador de la tarjeta de red se encuentra transmitiendo otros paquetes, el paquete quedará encolado para ser despachado según el orden en el que llegó [8].

Figura 2.4: Ejemplo de cola FIFO.

Los sistemas Linux, de manera predeterminada, tienen habilitado la disciplina de cola “PFIFO_FAST”, esta es una cola muy sencilla que consiste en tres bandas; la 0, 1 y 2. Cada una de las bandas es una prioridad, es decir, la banda 0 mantiene la prioridad más alta y la banda 2 la prioridad más baja. En cada banda los paquetes son despachados mediante la disciplina “FIFO”. Mientras la banda 0 se encuentre con paquetes encolados para enviar, no se dará lugar a las bandas 1 y 2 para que despachen los paquetes de sus respectivas colas, hasta que la banda 0 haya despachado completamente toda su cola.

Para ordenar los paquetes a una banda específica, se utiliza el campo “Tipo de Servicio” de la cabecera del protocolo Internet. El núcleo de Linux tiene establecido un mapa de prioridades para según el valor de este campo dirigir los paquetes a la banda que le corresponde [9].

Para observar las estadísticas de la disciplina de cola que se encuentre habilitada, se ha creado el siguiente código, que está constantemente receptando la información de la interfaz sobre la cual se aplicó la disciplina de cola.

- *#!/bin/bash*
- *watch -n0 '*
- *tc -s -d qdisc show dev wlan1;*
- *echo "=== Clases ====================";*
- *tc -s -d class show dev wlan1;'*
- *exit 0*

Cuando se tiene configurado alguna disciplina de cola y se desea volver a “pfifo_fast” para no aplicar calidad de servicio sobre los paquetes de datos, se tiene que ejecutar el siguiente comando, el cual elimina todas las disciplinas de colas configuradas.

- *tc qdisc del dev wlan1 root*

2.3.4 Disciplina de cola con clase

Existen otros algoritmos de disciplinas de colas que clasifican el tipo de tráfico en clases. Una clase, básicamente, establece una categoría de

tráfico, por ejemplo, al tráfico de video se le puede asignar una clase, otra clase puede contener el tráfico de datos, etc. En la Figura 2.5, se representa que antes de que a un flujo de paquetes se le defina una clase, tiene que pasar por un proceso de filtrado que es el encargado de relacionar el flujo de datos a la clase correspondiente, este comportamiento tiene HTB (Hierarchical Token Bucket) [10].

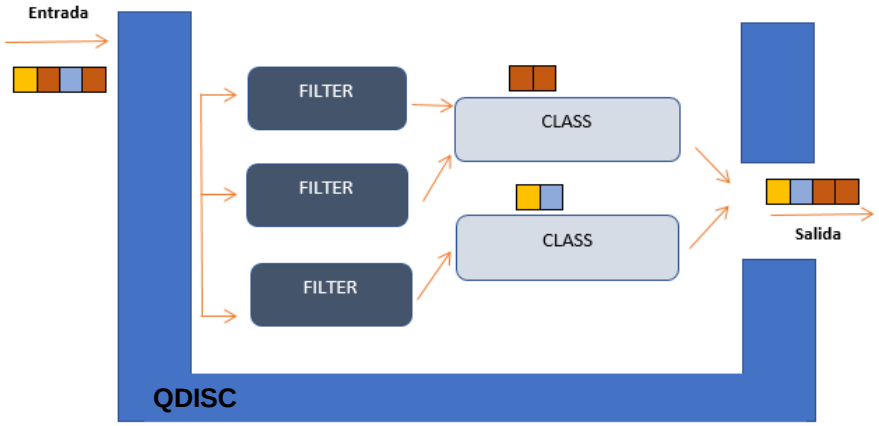


Figura 2.5: Ejemplo de cola con clases [10].

La disciplina de encolamiento “HTB” dispone de un conjunto de características para realizar una mejor gestión en la asignación de ancho de banda a determinado flujo de tráfico, también permite realizar priorización entre las clases brindando la posibilidad a una clase de tomar ancho de banda de otra si no lo está utilizando de acuerdo a los límites y prioridades establecidos.

La disciplina HTB (Hierarchical Token Bucket) trabaja de manera jerárquica tal como su nombre lo indica. En la Figura 2.6 se muestra que la parte más alta de la jerarquía va a tener la capacidad máxima que se dispone para la transferencia de datos, es desde allí donde se ramifican las clases hijas. Cuando se configura “HTB”, se define la capacidad asegurada de transferencia de la clase con el parámetro “rate”, para poder extender esta capacidad a su nivel máximo deseado se debe configurar el parámetro “ceil”, de esta manera puede solicitar ancho de banda a otras clases que no estén haciendo uso de su capacidad

asignada. Para la clase 1:1 se tiene 10 Mbps, a partir de esta clase se empiezan a ramificar las clases hijas y se les asigna el ancho de banda garantizado y el tope máximo a alcanzar, teniendo en consideración la prioridad asignada a cada clase.

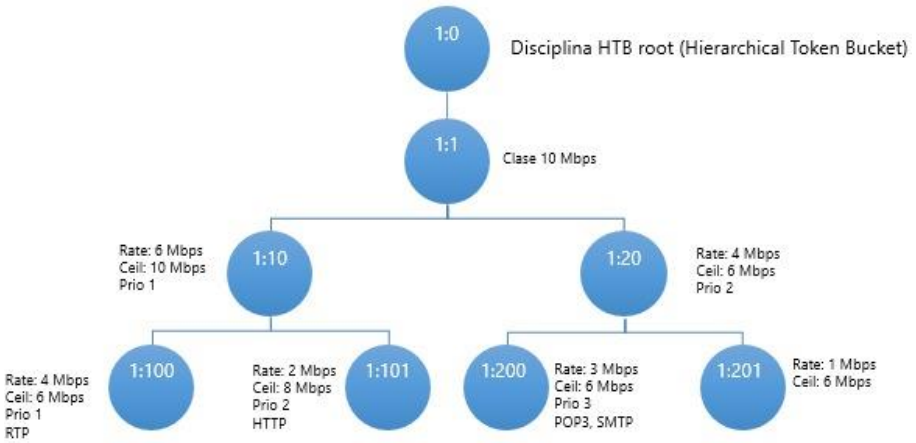


Figura 2.6: Diseño de árbol HTB [4].

2.3.5 Aplicando disciplina de cola HTB

Para aplicar la disciplina de cola HTB, se debe ejecutar el siguiente comando:

- *tc qdisc add dev wlan1 root handle 1: htb default 20*

Con el comando anterior se está asignando HTB a la interfaz wlan1, con el parámetro “default 20” se indica que todo paquete de dato que no sea marcado va a ser direccionado a la clase hija 1:20.

2.3.6 Creando árbol de cola HTB

Se define la clase padre con 10Mbps con el siguiente comando:

- *tc class add dev wlan1 parent 1: classid 1:1 htb rate 10000kbit ceil 10000kbit*

A continuación, se crea las clases hijas dividiendo los 10Mbps, en 6Mbps para la clase 1:10 con prioridad 1 y 4Mbps para la clase 1:20.

- *tc class add dev wlan1 parent 1:1 classid 1:10 htb rate 6000kbit ceil 10000kbit prio 1*

- *tc class add dev wlan1 parent 1:1 classid 1:20 htb rate 4000kbit ceil 10000kbit*

Se crean las subclases hijas de la clase hija 1:10, dividiéndolo en dos subclases de 4Mbps y 2Mbps con prioridad 1 y 2 respectivamente.

- *tc class add dev wlan1 parent 1:10 classid 1:100 htb rate 4000kbit ceil 6000kbit prio 1*
- *tc class add dev wlan1 parent 1:10 classid 1:101 htb rate 2000kbit ceil 8000kbit prio 2*

Se crean las subclases hijas de la clase hija 1:20, dividiéndolo en dos subclases de 3Mbps con prioridad 3 y 1Mbps.

- *tc class add dev wlan1 parent 1:20 classid 1:200 htb rate 3000kbit ceil 6000kbit prio 3*
- *tc class add dev wlan1 parent 1:20 classid 1:201 htb rate 1000kbit ceil 6000kbit*

2.3.7 Aplicando filtros

Con los siguientes comandos se relacionan las etiquetas y los filtros para las clases hijas que se han creado, en donde básicamente el parámetro “handle 100” especifica la marca asignada anteriormente a los paquetes con la etiqueta 100 y el parámetro “fw classid 1:100” especifica la clase hija a la que será direccionado. Esto permitirá priorizar el tráfico de acuerdo a la jerarquía establecida en “HTB”.

- *tc filter add dev wlan1 protocol ip parent 1: handle 100 fw classid 1:100*
- *tc filter add dev wlan1 protocol ip parent 1: handle 101 fw classid 1:101*
- *tc filter add dev wlan1 protocol ip parent 1: handle 200 fw classid 1:200*

Una vez que se han nombrado las herramientas de “software libre” con las cuales se van a realizar las pruebas de laboratorio, es necesario indicar que las configuraciones se realizan sobre un equipo con el rol de enrutador de borde. En la Figura 2.7, se puede observar el escenario propuesto de una red con máscara de 24 bits, es decir que se dispondrán como máximo de 254 equipos en ambos segmentos de red, los cuales

podrán generar todo tipo de tráfico de paquetes de datos y este debe ser considerado y evaluado antes de ser transmitido a la red de Internet.

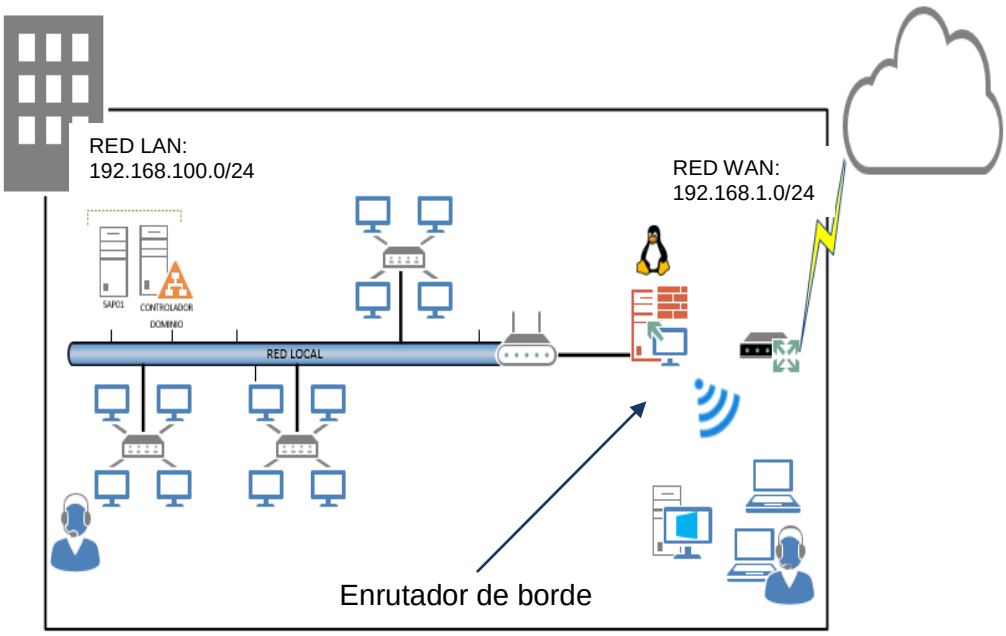


Figura 2.7: Infraestructura de red, con enrutador de borde aplicable calidad de servicio.

2.3.8 Herramientas de monitoreo

Como prerrequisito se debe instalar “Apache”, la instalación de “SmokePing” se lo realiza con el siguiente comando apt-get install smokeping, los archivos de configuración son los siguientes

- */etc/smokeping/config.d/Alerts*
- */etc/smokeping/config.d/Database*
- */etc/smokeing/config.d/General*
- */etc/smokeping/config.d/pathnames*
- */etc/smokeping/config.d/Presentation*
- */etc/smokeping/config.d/Probes*
- */etc/smokeping/config.d/Slaves*
- */etc/smokeping/config.d/Targets*

Para efecto de este proyecto se ha realizado la configuración del siguiente archivo /etc/smokeing/config.d/General. Primero se entra en modo de edición del fichero con el siguiente comando:

- *nano /etc/smokeping/config.d/General*

Una vez dentro del mismo se debe configurar los siguientes parámetros:

- *owner = Pablo Lopez // Nombre de propietario*
- *contact = losipablo@hotmail.com // se enviarán alertas*
- *mailhost = mail.outlook.com // servidor de correo*
- *# NOTE: do not put the Image Cache below cgi-bin*
- *# since all files under cgi-bin will be executed ... this is not*
- *# good for images.*
- *cgiurl = http://192.168.1.50/cgi-bin/smokeping.cgi //acceso vía web*
- *# specify this to get syslog logging*
- *syslogfacility = local0*
- *# each probe is now run in its own process*
- *# disable this to revert to the old behaviour*
- *# concurrentprobes = no*
-
- *@include /etc/smokeping/config.d/pathnames*

Otra de las herramientas de monitoreo que se debe instalar es MRTG, a fin de monitorear el tráfico de datos en un dispositivo de la red.

Ejecutar el siguiente comando para iniciar con la instalación.

- *$sudo apt-get install mrtg*

Se necesita crear un archivo de configuración, para lo cual, se debe ejecutar la siguiente sentencia.

- *cfgmaker public@192.168.1.1 >> /etc/mrtg.cfg*

La sentencia anterior crea un archivo, pero se cargan las configuraciones de manera automática, entonces es necesario que se modifique el archivo creado, estableciendo los parámetros personalizados que apliquen al equipo en el cual se está instalando. Se debe editar el archivo /etc/mrtg.cfg.

- *$ nano /etc/mrtg.cfg*

Una vez dentro del archivo a modificar, se puede apreciar la siguiente configuración que se creó de manera automática.

- *### Global Config Options*
- *# for Debian*
- *WorkDir: /var/www/mrtg*
  Language: spanish
  RunAsDaemon: yes
  Interval: 5

En la línea del parámetro “WorkDir”, se debe definir la ruta donde se van a publicar los gráficos y el servidor web debe tener acceso a la ruta para poder visualizar los gráficos vía web.

La opción de “Language” permite establecer el idioma en el cual se mostrarán los datos.

Si se desea que se inicie la aplicación de MRTG de manera automática, se debe definir el parámetro “RunAsDaemon” en “yes”. La opción anterior debe ir acompañada de “Interval” que es donde se define cada “cuantos minutos” se va a ejecutar MRTG.

- *### DATOS DE TRAFICO ###*
- *Target[pls.local]: 2:public@192.168.1.1:*
- *PageTop[pls.local]: <h1>Uso de Red</h1>*
- *Options[pls.local]: growright, bits*
- *Title[pls.local]: Datos de Red*

La opción de “Target”, es la sentencia en la cual se define el interfaz y el dispositivo a monitorear, donde:

- **Target[nombre],** se define un nombre como etiqueta
- **:2,** se indica el número de la interfaz que se va a monitorear
- **public,** es el método de autenticación para la lectura de datos que usa SNMP, la instalación de SNMP se la muestra en la sección de Anexos.
- **192.168.1.1,** es la ip del dispositivo que se va a monitorear.

La opción de “PageTop”, sirve para agregar texto en la parte superior de la página html de los datos a presentarse.

Si se desea cambiar la manera gráfica de presentar los datos se puede usar el parámetro “Options”. En este caso con la sentencia “growright, bits” se está indicando que las gráficas van a aparecer de izquierda a derecha y se ha cambiado la unidad a bit, de manera predeterminada aparecen los gráficos de derecha a izquierda y los representa en bytes.

2.4 Plan de Implementación

En la Figura 2.8, se muestra cada fase con su respectiva duración, además se describen las actividades que se deben realizar, a fin de llevar a cabo este proyecto.

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
Medición de tipos de retardos	**21,13 días**	**mar 15/08/17**	**mié 13/09/17**	
Inicio	**0,5 días**	**mar 15/08/17**	**mar 15/08/17**	
Reunión de apertura	4 horas	mar 15/08/17	mar 15/08/17	
Planeación	**2,75 días**	**mar 15/08/17**	**jue 17/08/17**	
Levantamiento de información	2 días	mar 15/08/17	jue 17/08/17	3
Entregable: Documento de requerimientos	2 horas	jue 17/08/17	jue 17/08/17	5
Elaborar Plan de proyecto	2 horas	mar 15/08/17	mar 15/08/17	
Ejecución	**15 días**	**vie 18/08/17**	**vie 08/09/17**	
Análisis y Arquitectura	2 días	vie 18/08/17	mar 22/08/17	
Entregable: Documento de arquitectura	3 horas	mar 22/08/17	mar 22/08/17	9
Configuración de herramientas	10 días	mar 22/08/17	mar 05/09/17	9
Pruebas/Demos	3 días	mar 05/09/17	vie 08/09/17	11
Capacitación	**1,13 días**	**lun 11/09/17**	**mar 12/09/17**	
Transferencia de conocimiento	5 horas	lun 11/09/17	lun 11/09/17	12
Entrega de manuales	2 horas	lun 11/09/17	lun 11/09/17	
Cierre	**0,13 días**	**mié 13/09/17**	**mié 13/09/17**	
Reunión de cierre	1 hora	mié 13/09/17	mié 13/09/17	15

Figura 2.8: Plan de Implementación.

En la Figura 2.9, se muestra el Diagrama de Gantt donde de manera gráfica se expone el tiempo previsto para las tareas a ejecutarse.

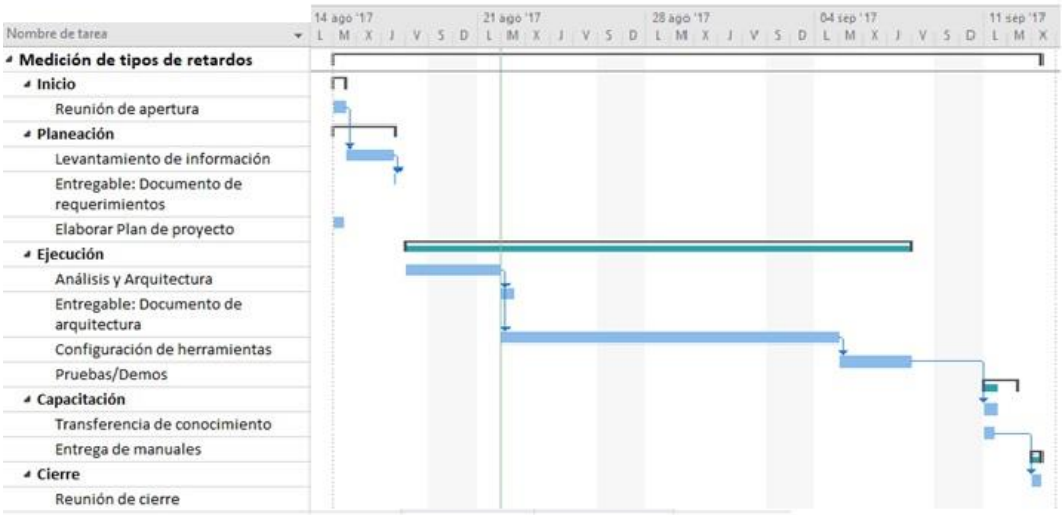


Figura 2.9: Diagrama de Gantt.

CAPÍTULO 3

3. INTERPRETACIÓN DE RESULTADOS.

En el presente capítulo se detallan los resultados obtenidos en la medición del retardo de las técnicas de calidad de servicio. Es relevante indicar que antes de realizar las respectivas configuraciones en los dispositivos para aplicar un método calidad de servicio a los paquetes de datos de interés, se debe tener en consideración la capacidad máxima del ancho de banda, a fin de establecer políticas de control.

Para realizar una evaluación objetiva, se elabora el escenario de pruebas para validar las disciplinas escogidas y determinar cuál es más eficiente al momento de ofrecer un tratamiento diferenciado a los paquetes de datos. Tal como se muestra la tabla 3 en donde se especifican los parámetros para las pruebas de las disciplinas a evaluar.

Equipo origen LAN	Equipo destino WAN	Transferencia	Canal de voz
Linux x86 192.168.100.101 100Mbps	Wndows 7 x64 192.168.1.110 54Mbps	Archivo ISO de 2GB	
Wndows 7 x86 192.168.100.110 100Mbps	Wndows 7 x64 192.168.1.110 54Mbps	Archivo ISO de 2GB	
Android 192.168.100.150			1 canal en G711a 90Kbps
Android 192.168.1.150			1 canal en G711a 90Kbps

Tabla 3: Parámetros de pruebas.

3.1 Resultados

Con el objetivo de conocer el método más apropiado para el tráfico de interés, a continuación, se presentan los resultados obtenidos en el ambiente preparado con las herramientas de software libre.

Al habilitar el servicio de enrutamiento en la distribución Linux Ubuntu Server 16.04, la disciplina de cola que viene configurada de manera predeterminada es “PFifo_Fast”. En la Figura 3.1, se muestra la estadística de cada banda de la disciplina de cola antes mencionada, donde al tráfico interactivo se le da mayor prioridad enviándolo a la banda 1.

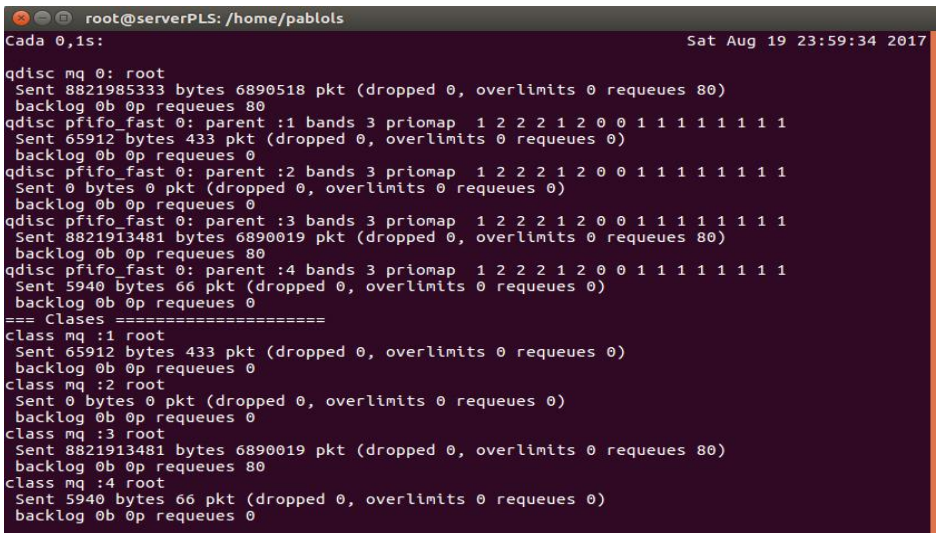


Figura 3.1: Estadística de “PFifo_Fast”.

Como se describió en capítulos anteriores, con “SmokePing” se puede obtener estadísticas del retardo promedio de los paquetes así como también la variación de estos retardos. A continuación, en la Figura 3.2 se puede apreciar líneas cortas en posición horizontal de diferentes colores que representan la pérdida de paquetes y el retardo promedio en milisegundos de un determinado tiempo. Además, se puede ver que la disciplina de cola “PFifo_Fast” aproximadamente a las 23H45, tiene el promedio de retardo más alto encontrándose entre 140 y 160 milisegundos. En el área del gráfico se puede observar barras verticales de color gris, cuando el color es tenue indica que el tiempo de variación de retardo entre paquetes de datos “jitter” es menor, por el contrario, cuando el color es oscuro significa que el tiempo de variación de retardo entre paquetes se elevó.

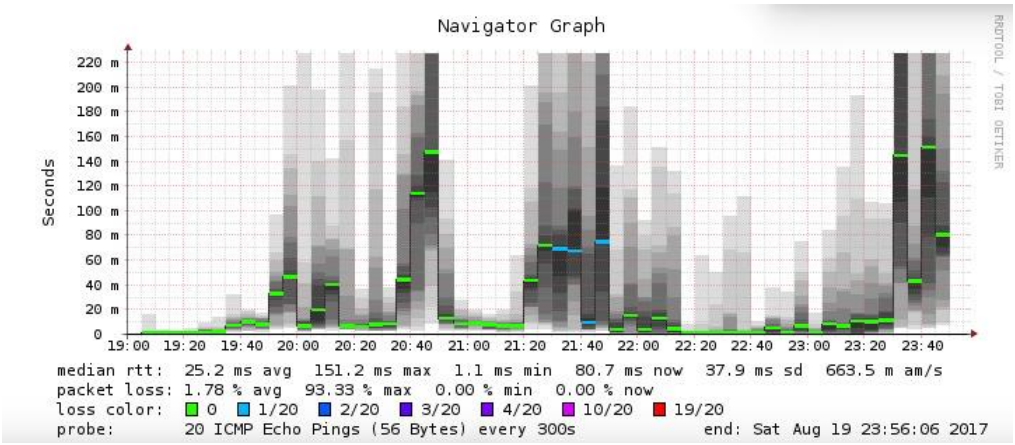


Figura 3.2: Retardo promedio con SmokePing – pfifo_fast.

A fin de poder visualizar la carga de tráfico se ha utilizado “MRTG” (Multi Router Traffic Grapher). En la Figura 3.3, se representa que con la disciplina de cola “PFifo_Fast” no existe un control en el uso del ancho de banda y se valida que la carga de tráfico se eleva.

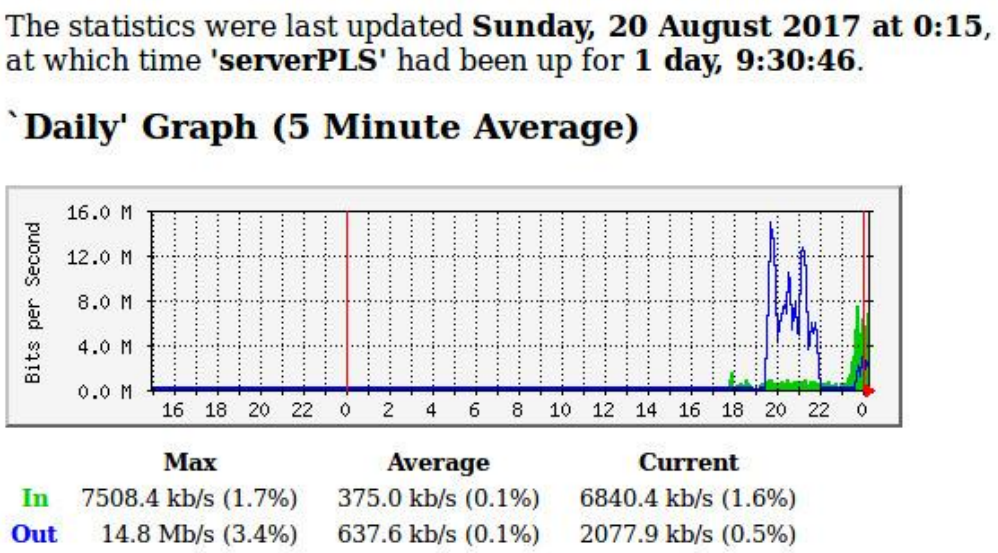


Figura 3.3: Monitoreo de carga de tráfico.

Por otra parte, se realizó el monitoreo de retardo aplicando una disciplina de cola diferente. Esta disciplina de cola llamada “HTB” (Hierarchical Token Buckets), permite configurar de manera granular la asignación del ancho de banda para determinados tipos de servicios. En la Figura 3.4, se muestra que el

tiempo promedio de retardo más alto aproximadamente a las 19H55, se encuentra entre 30 y 40 milisegundos, determinando que esta disciplina de cola aplica un control en el tráfico.

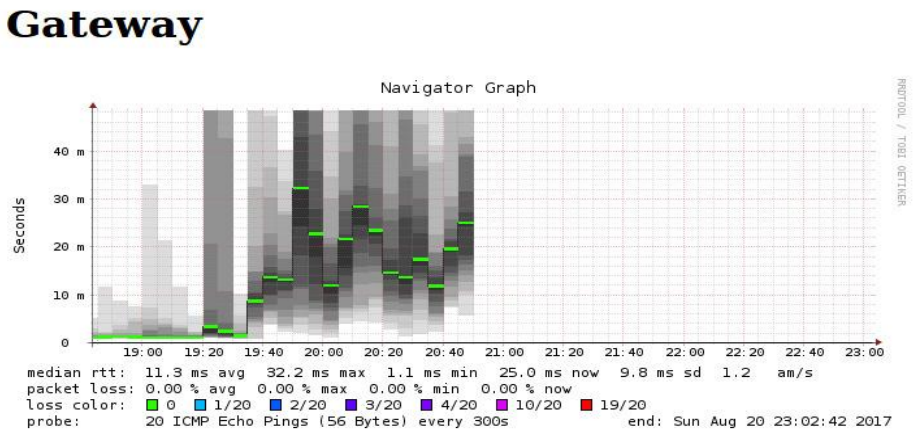


Figura 3.4: Retardo promedio con "SmokePing" – "HTB".

La herramienta de monitoreo "IPTraf" permite obtener estadística de la interfaz de red. En la Figura 3.5 se muestra que se clasifica el resultado por el tipo de protocolo, mostrando información de la cantidad de paquetes y de bytes enviados y recibidos. También se aprecia la tasa de transferencia entrante y saliente que ha sido controlada por la política aplicada con el control de tráfico "TC".

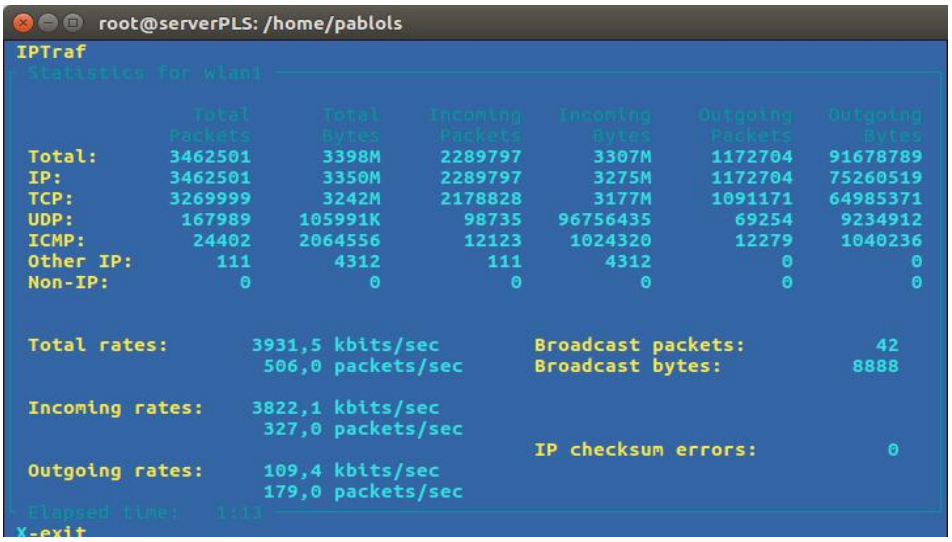


Figura 3.5: Estadísticas de Interfaz de red.

Como se muestra en la Figura 3.6, con “HTB” se puede mantener la carga de tráfico controlada. Se aprecia claramente que las líneas donde se grafican los datos no se elevan. Por el contrario, se visualiza que con “PFifo_Fast” la carga de tráfico se eleva, debido a que no se dispone de un control en el uso del ancho de banda.

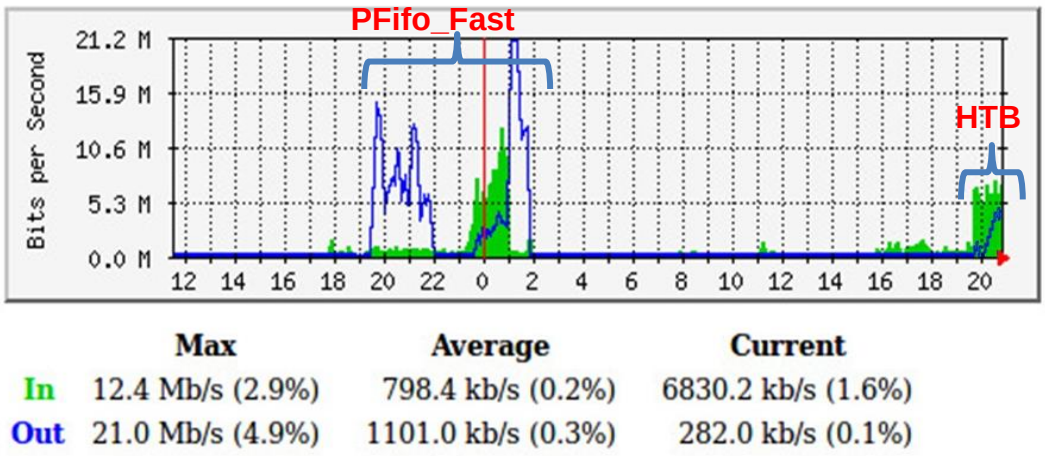


Figura 3.6: Comparación en la carga de tráfico.

Con base a las configuraciones de clases realizadas con “HTB” descritas en el capítulo 2, en la Figura 3.7, se muestra la estadística del tráfico de datos que ha pasado por las diferentes clases creadas; el parámetro “lended” indica los bytes que una clase padre ha prestado a las clases hijas y el parámetro “borrowed” indica la cantidad de bytes que la clase hija ha pedido prestado a la clase padre, a fin de ampliar la capacidad máxima en el uso del ancho de banda.

```
Cada 0,1s:

qdisc htb 1: root refcnt 5 r2q 10 default 20 direct_packets_stat 1253609 ver 3.17 direct_qlen 1000
 Sent 107806715 bytes 1330827 pkt (dropped 0, overlimits 1658 requeues 5)
 backlog 0b 0p requeues 5
=== Clases ====================
class htb 1:101 parent 1:10 prio 2 quantum 25000 rate 2Mbit ceil 8Mbit linklayer ethernet burst 1600b/1
b overhead 0b level 0
 Sent 9229218 bytes 61027 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
 lended: 59938 borrowed: 839 giants: 0
 tokens: 95875 ctokens: 23968

class htb 1:100 parent 1:10 prio 1 quantum 50000 rate 4Mbit ceil 6Mbit linklayer ethernet burst 1600b/1
b overhead 0b level 0
 Sent 1717207 bytes 16191 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
 lended: 16191 borrowed: 0 giants: 0
 tokens: 46937 ctokens: 31286

class htb 1:10 parent 1:1 rate 6Mbit ceil 10Mbit linklayer ethernet burst 1599b/1 mpu 0b overhead 0b cb
 Sent 10946425 bytes 77218 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
 lended: 540 borrowed: 299 giants: 0
 tokens: 31286 ctokens: 18775

class htb 1:1 root rate 10Mbit ceil 10Mbit linklayer ethernet burst 1600b/1 mpu 0b overhead 0b cburst 1
 Sent 10946425 bytes 77218 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
 lended: 299 borrowed: 0 giants: 0
 tokens: 18775 ctokens: 18775
```

Figura 3.7: Estadísticas de clases “HTB”.

Cabe destacar que con SmokePing, se ha realizado una comparación entre las disciplinas de cola “PFifo_Fast” y “HTB”. En la Figura 3.8, se compara el retardo promedio con ambas disciplinas, donde claramente se aprecia que el retardo con “PFifo_Fast” se mantiene en tiempos elevados, pero al configurarse una disciplina como HTB, este tiempo de retardo promedio decrece notablemente.

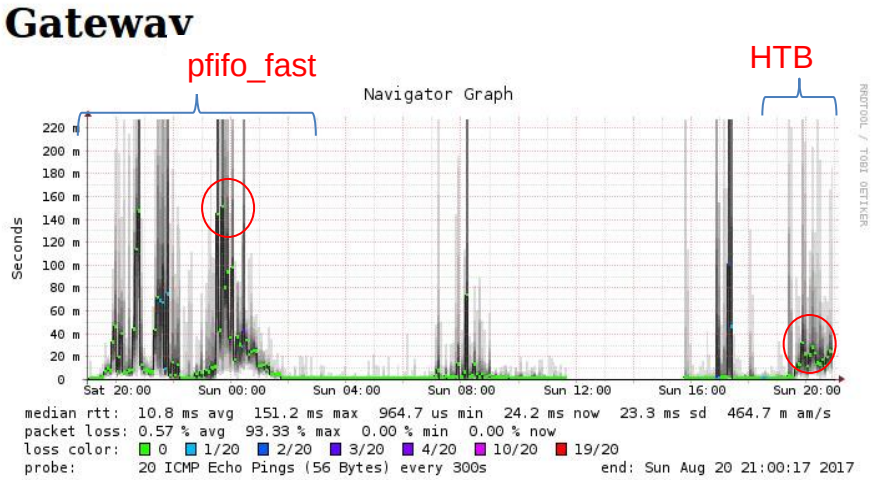


Figura 3.8: Comparación de retardo promedio.

En la tabla 4, se resumen los resultados obtenidos con las disciplinas evaluadas en el laboratorio, se puede observar que al aplicar una disciplina de colas con clase como “HTB” se consigue controlar la capacidad del enlace, permitiendo que las diferentes variedades de tráfico cuenten con una parte asegurada de dicha capacidad. De esta manera se alcanza controlar el retardo para el tráfico de interés para que los servicios no se vean afectados en situaciones de mayor demanda y con esto contar con la calidad de servicio esperada.

Disciplina de Cola	Transferencia Máxima	Retardo promedio	Retardo máximo/mínimo	Calidad de voz
Fifo fast	22 Mbps	155 ms	500 ms / 50 ms	Robotizada y entrecortada
HTB	6 Mbps	40 ms	50 ms / 30 ms	Buena calidad

Tabla 4: Resultado de pruebas.

CONCLUSIONES Y RECOMENDACIONES.

Con el diseño propuesto se identifican herramientas como “SmokePing”, que disponen de funcionalidades que operan en tiempo real para alertar al administrador de la red, con el fin de tomar decisiones proactivas para evitar la ocurrencia de un evento que perjudique el tráfico de interés.

Con la implementación realizada, se comprobó que la aplicación de un método de calidad de servicio adecuado como “HTB” permite priorizar la entrega de datos de interés de manera más eficiente que otro método como “FIFO FAST”. Además, de realizar un control del uso del ancho de banda por medio de políticas establecidas con el fin de entregar los diferentes flujos de datos y no solo el tráfico interactivo al que “FIFO FAST” asigna prioridad dejando en cola de espera el resto de datos.

En síntesis, se comprobó que se puede mejorar la calidad de servicio, estableciendo un método adecuado para clasificar los paquetes de datos y despacharlos según el tratamiento diferenciado que se requiera y para esto “HTB” se constituye como una herramienta poderosa que brinda un tratamiento granular para lograr disponer de servicios de calidad según el tráfico de interés dependiendo la línea de negocio de cada empresa.

Se debe considerar que antes de implementar un método de calidad de servicio, es importante entender el modelo de negocio, a fin de determinar el tipo de tráfico en el cual se desea implementar la diferenciación y priorización.

Se recomienda identificar el tráfico que se considera de interés para la empresa, a fin de implementar calidad de servicio apropiadamente. También es recomendable contratar un servicio de internet que por medio de “SLA” garantice la diferenciación y priorización de los datos de interés de salida hacia la Internet.

Para definir el ancho de banda mínimo a un determinado servicio, primero se debe recurrir a los detalles técnicos del fabricante, por ejemplo si se desea conocer los requerimientos mínimos de red para un servicio de mensajería instantánea como “Skype Empresarial”, se puede hacer uso de la calculadora “Skype for Business BW Calc - Version 2 70” creada por el mismo fabricante, donde básicamente se

ingresa información como el número de usuarios que harán uso del servicio y el tipo de uso que se le dará a la herramienta.

BIBLIOGRAFÍA

[1] Microsoft (2003, Marzo 28). How TCP/IP Works – TCP/IP Protocol Architecture [Online]. Disponible en: https://technet.microsoft.com/en-us/library/cc786128(v=ws.10).aspx#w2k3tr_tcpip_how_ejod

[2] Juan C. Martínez (2010, Marzo 08). Calidad de Servicio [Online]. Disponible en: http://cic.puj.edu.co/wiki/lib/exe/fetch.php?media=materias:daysenr:daysenr_-_calidad_de_servicio_qos_.pdf

[3] Chris Lewis (2006, Mayo 26). Implementing Quality of Service Over Cisco MPLS VPNs [Online]. Disponible en: http://www.ciscopress.com/articles/article.asp?p=471096&seqNum=6

[4] Universidad de Sevilla (2017, agosto 01). Filtrado de paquetes con Netfilter [Online]. Disponible en: https://www.dte.us.es/docencia/etsii/gii-ti/tecnologias-avanzadas-de-la-informacion/Laboratorio-2-Netfilter.pdf

[5] MicroTeknologías (2009, Octubre 20). IPtables, un cortafuegos en el núcleo de Linux [Online]. Disponible en: https://microteknologias.wordpress.com/2009/10/20/iptables-un-cortafuegos-en-el-nucleo-de-linux/

[6] Daniel Morató Osés (2016, febrero 16). Nuevos Servicios de Red en Internet [Online]. Disponible en: https://www.tlm.unavarra.es/~daniel/docencia/nsri/nsri11_12/slides/09-DiffServ.pdf

[7] Angrisani, Ventre, Peluso y Tedesco “Measurement of Processing and Queuing Delays Introduced by an Open-Source Router in a Single-Hop Network,” IEEE Transactions on Instrumentation and Measurement, vol. 55, no. 4, Agosto 2006

[8] Eva Castro. Control de tráfico y DiffServ en Linux [Online]. Disponible en: https://evacastro.gitbooks.io/internet/content/control_de_trafico.html

[9] Martin A. Brown. Components of Linux Traffic Control [Online] Disponible en: http://tldp.org/en/Traffic-Control-HOWTO/ar01s04.html#c-txqueuelen

[10] Yaron Benita. Kernel Korner – Analysis of the HTB Queuing Discipline [Online]. Disponible es: http://www.linuxjournal.com/article/7562

ANEXOS

a. GLOSARIO

MRTG – Multi Router Traffic Grapher

WAN – Wide Area Network

LAN – Local Area Network

D-ITG – Distributed Internet Traffic Generator

TCSIM – Traffic Simulator

IPv4 – Internet Protocol version 4

IPv6 – Internet Protocol version 6

TTL – Time to Live

ToS – Type of Service

TC – Traffic Control

QDISC – Queuing Disciplines

FIFO – First In First Out

HTB – Hierarchical Token Buckets

TCP/IP – Transmission Control Protocol/Internet Protocol

WLAN – Wireless Local Area Network

Mbps – Megabit per second

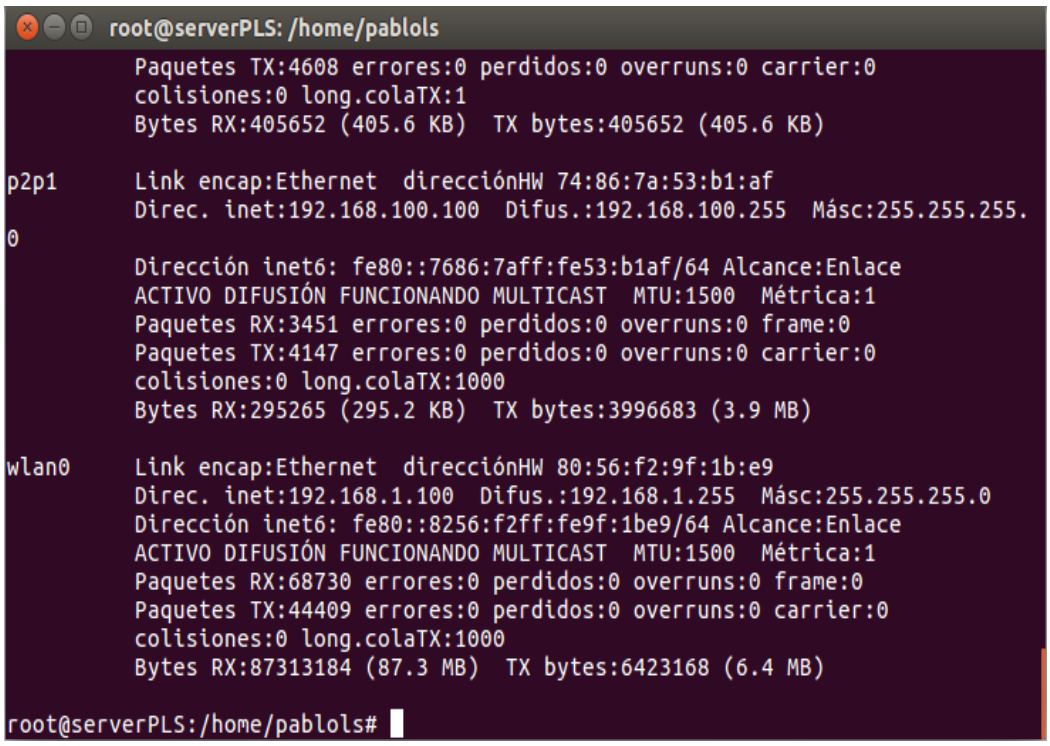
SNMP – Simple Network Management Protocol

SLA – Service Level Agreement

b. CONFIGURACIÓN D-ITG

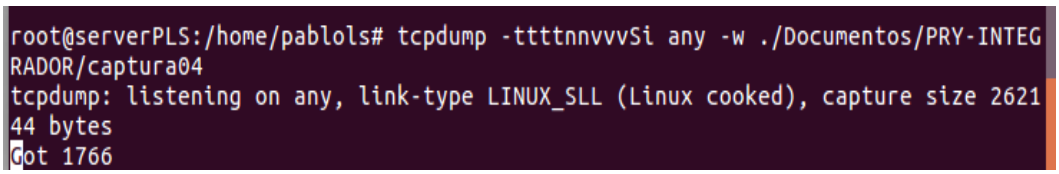
1. Pruebas de simulación de tráfico

Antes de iniciar con las configuraciones para las pruebas de simulación de tráfico, se deben configurar las interfaces de red con IPs estáticas para llevar un mejor control en la asignación de IPs. En la Figura “Configuración de las interfaces de red”, se evidencia la configuración.



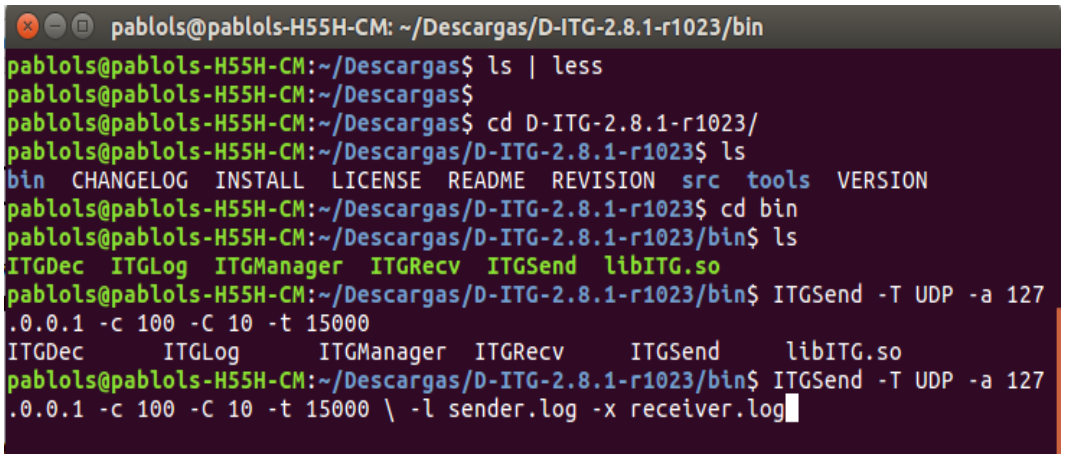
Configuración de las interfaces de red

La herramienta tcpdump, sirve para obtener estadística de los paquetes transmitidos y recibidos. A continuación en la Figura “Captura de información con tcpdump”, se aprecian los resultados.



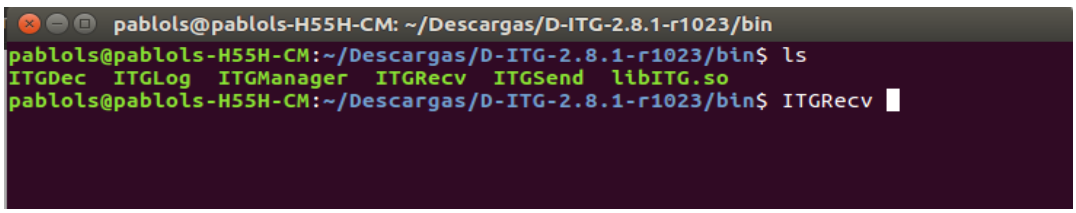
Captura de información con “tcpdump”.

La herramienta de simulación de tráfico D-ITG, ayuda a generar diferentes tipos de tráficos y la cantidad de bytes que se desea enviar. En el emisor se debe ejecutar la sentencia que se muestra en la Figura “Simulación de tráfico con D-ITG”, donde se está enviando tráfico UDP. Con los parámetros -l y -x se define el nombre del archivo de registros que se creará en el emisor y receptor respectivamente.



Simulación de tráfico con D-ITG

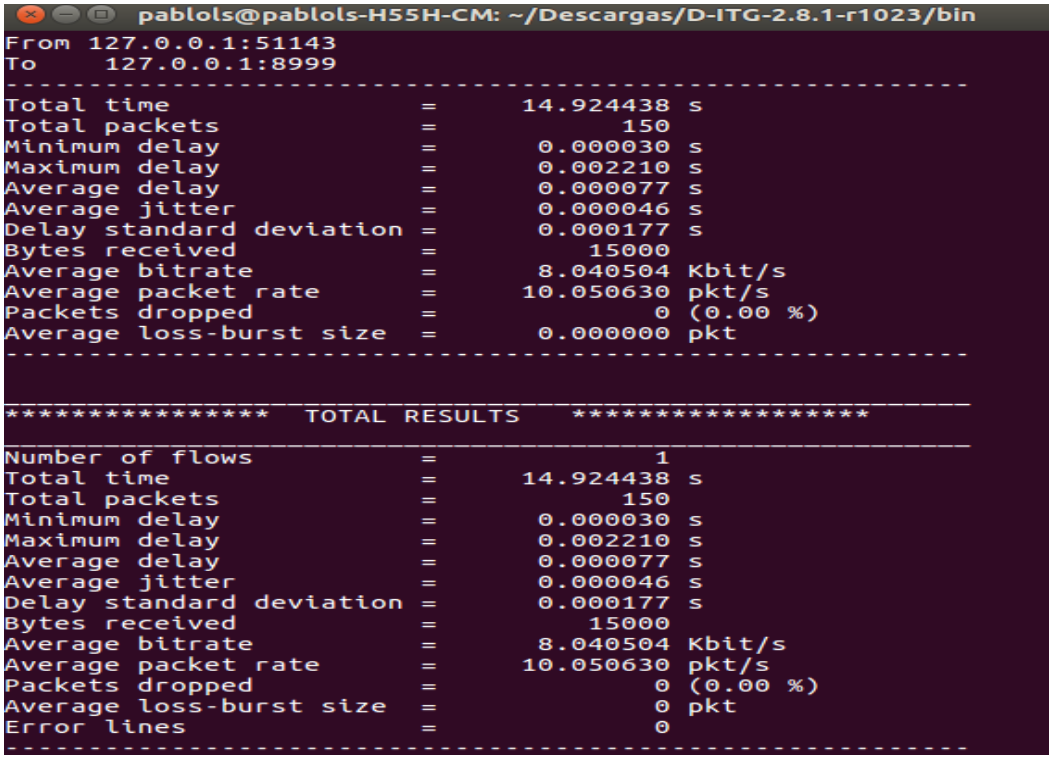
Una vez que en el emisor se ha configurado los parámetros de D-ITG, se debe ejecutar en el receptor el comando ITGRecv, donde empezará a sensar alguna conexión por parte de ITGSend. A continuación, en la Figura “Recepción de tráfico con D-ITG” se evidencia el comando.



Recepción de tráfico con D-ITG

Para que los archivos que se generan tanto en el emisor como en el receptor sean leídos, se tiene que ejecutar el comando “ITGDec” seguido del nombre del archivo generado, a fin de mostrar los resultados por medio de consola. En la

Figura “Estadística de los paquetes generados en la simulación” se aprecia el uso del comando “ITGDec”



Estadística de los paquetes generados en la simulación

c. CONFIGURACIÓN SNMP

2. Pasos de instalación de SNMP

Dependiendo de la distribución de Linux se debe ejecutar el comando correspondiente para la instalación de SNMP, que es un protocolo que se encarga de recolectar información de la red. Para la instalación correspondiente se debe ejecutar la siguiente sentencia.

$sudo apt-get install snmpd

Luego se debe editar el fichero de “snmpd.conf” que se encuentra en la siguiente ruta.

$ nano /etc/snmp/snmpd.conf

Para activar un parámetro se debe eliminar el símbolo “#”, que está al inicio de la línea. Esto es para permitir que MRTG pueda hacer lectura de los datos recolectados por SNMP.

com2sec readonly default public

Se procede a guardar los cambios y se reinicia el servicio de snmp con la siguiente sentencia.

$ /etc/init.d/snmpd/ restart