CALIDAD DE SERVICIO IPv6

Laura Vanessa Cabrera Sanmartín

Documento presentado como opción de grado de ingeniería electrónica en la facultad de ingenierías de la Universidad Tecnológica de Bolívar

Cartagena de Indias D. T. y C. ***2012***

Universidad Tecnológica de Bolívar
CARTAGENA DE INDIAS

CALIDAD DE SERVICIO EN IPV6

LAURA VANESSA CABRERA SANMARTÍN

Asesor:

M. Sc. RICARDO JAVIER ARJONA ANGARITA

MINOR EN TELECOMUNICACIONES

FACULTAD DE INGENIERÍAS

PROGRAMA DE INGENIERÍA ELECTRÓNICA

CARTAGENA DE INDIAS D. T. Y C.

2012

TABLA DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCIÓN

Durante los últimos años, el crecimiento exponencial de la Internet y el agotamiento del espacio de direcciones ofrecidas por el actual protocolo de red IPv4, ha conllevado a la evolución del mismo. El desarrollo del protocolo de internet de nueva generación, ha venido llevándose a cabo de manera paralela al aprovechamiento de las direcciones ofrecidas actualmente por IPv4. Surgió el conjunto de protocolos y estándares, IPv6 como solución a los problemas presentes con IPv4, y que incluye los conceptos de metodologías propuestas para una actualización del protocolo actual.

Es importante demarcar las diferencias que se presentaran con la implementación de un nuevo protocolo de internet, ¿Qué mejoras trae consigo? ¿Qué tipo de servicios se podrá ofrecer? ¿Qué tan eficiente se realizará la transmisión de paquetes? ¿Traería alguna desventaja?

A raíz de la implementación del protocolo IP de nueva generación, es necesario conocer las características que lo distinguen del anterior protocolo, para estar en la capacidad de manejarlo y de realizar la transición de uno a otro en la arquitectura de red existente.

Otro factor muy importante que se ha venido desarrollando hace mucho tiempo, es la calidad de servicio implementada en las redes. Qué tratamiento deben recibir ciertos tipos de tráfico, como se ven estos afectados por factores de transmisión de paquetes dentro de la red, y por los protocolos de enrutamiento.

Deben estudiarse las diferentes metodologías de encolamiento y manipulación de tráfico, así como las técnicas de evasión de congestión, para poder realizar una buena asignación de recursos

a los tráficos que lo necesiten, y que si se presenta algún error, el usuario no se vea afectado. Cumpliendo siempre los niveles de calidad de servicio que un cliente contrate.

Los parámetros de calidad de servicio QoS, estos últimos determinan el comportamiento de la red, con respecto a características de servicio definidas, los cuales pueden ser determinados con las siguientes medidas:

- Ancho de banda

- Retardo de transporte

- Jitter

- Pérdida de paquetes.

Es necesario definir esquemas de calidad de servicio, donde se definan claramente cuales son los acuerdos de nivel de servicio que serán ofrecidos, y que tratamiento se le dará a cada tipo de tráfico. En este esquema también debe escogerse la arquitectura sobre la cual trabajará el esquema.

Con el desarrollo de este documento se busca apreciar los cambios presentes en los parámetros de calidad de servicio QoS dentro del nuevo protocolo de redes IPv6, para luego realizar una comparación con respecto a las redes actualmente implementadas.

Se podrá apreciar que el desarrollo que hasta el momento se ha llevado a cabo, en cuestiones de QoS, es bastante robusto, por lo que se implementaron características dentro del protocolo IPv6 que aprovechen al máximo lo desarrollado. Es incluso más sencillo.

1. PROTOCOLO IPv6

Al vislumbrarse el posible agotamiento de los bloques de direcciones IP, debido al crecimiento tecnológico; se inició con la investigación y el desarrollo de un nuevo protocolo de internet para remplazar el implementado actualmente.

Un grupo de trabajo en la IETF (*Internet Engineering Task Force)*, se encargó de presentar las características que debía tener el nuevo protocolo, y fue publicada oficialmente la versión del protocolo IPv6, en 1990. Este mantiene los principales conceptos del protocolo, descartando las características muy poco utilizadas en la práctica; y agregando nuevas características como metodología de solución a los problemas que se presentan en el protocolo IPv4.

1.1. CARACTERÍSTICAS GENERALES

El motivo principal por el cual fue desarrollado un nuevo protocolo de internet, fue la cantidad de direcciones; por consiguiente, el tamaño de una dirección IPv6 aumenta de 32 a 128 bit, que equivale aproximadamente a $3{,}4 \cdot 10^{38}$ direcciones disponibles; asegurando una dirección pública a cada dispositivo conectado a una red.

Se cambió el formato de la cabecera. Tiene un mayor tamaño que en IPv4, pero con menos campos de trabajo, lo que ocasiona la manipulación más eficiente de los paquetes. La Figura 1.1 muestra el formato de la cabecera IPv4. Los campos en color naranja desaparecen totalmente en la cabecera IPv6, los campos en color verde tienen homólogos en la cabecera IPv6, pero no desaparecen.

Se incorpora un mecanismo de autoconfiguración de direcciones, a través del cual los nodos tienen la capacidad de auto asignarse una dirección IPv6 sin intervención del usuario.

También se incorpora un protocolo de interacción con vecinos, con el cual se prescinde del uso de los protocolos ARP y “Router Discovery” de IPv4. Eliminando el uso de los mensajes broadcast.

Figura 1.1. Formato cabecera IPv4

1.2. PAQUETE IPV6

El tamaño de la cabecera de un paquete IPv6 es de 40 byte, que es el doble del tamaño de la cabecera IPv4. Cambio que se dio debido al aumento del tamaño de las direcciones, de 32 a 128 bit. Adicionalmente presenta una reserva de 40 bytes de cabeceras adicionales, para futuras expansiones del funcionamiento de IPv6. Estas cabeceras se ubican después de la cabecera IPv6 y antes de la cabecera de protocolo superior (UDP o TCP).

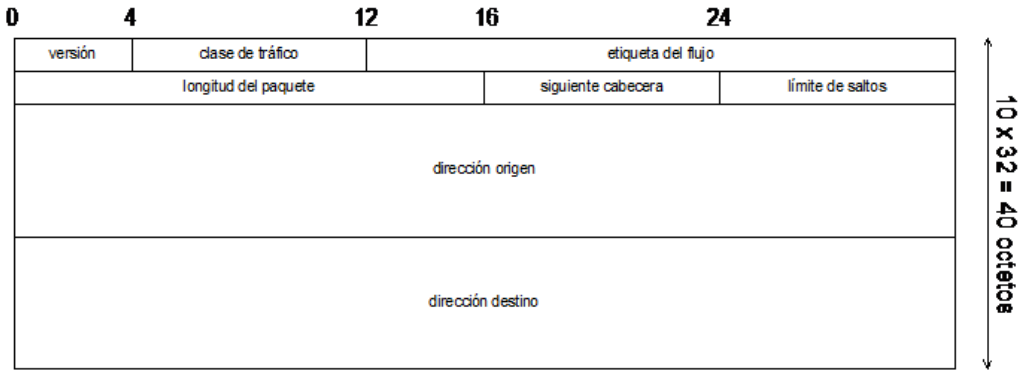

Figura 1.2. Formato de un paquete IPv6

Tabla 1.1. Campos de la cabecera IPv6

Campo IPv6	Tamaño [bits]	Descripción
Versión	4	Indica la versión del protocolo IP, para que sea interpretada por los enrutadores para procesar el paquete.
Clase de tráfico	8	Muestra diferentes clases o prioridades de los paquetes IPv6, para que los enrutadores puedan clasificar los paquetes según el tipo de tráfico al que pertenecen.
Etiqueta de flujo	20	Permite que los enrutadores identifiquen los paquetes que deben tratarse igualmente, a través de la distinción de flujo de paquetes.
Longitud de paquete	16	Indica el tamaño de la carga útil del paquete.
Siguiente cabecera	8	Identifica el tipo de cabecera inmediatamente continua a la cabecera del presente paquete IPv6.
Límite de saltos	8	Máximo número de saltos que puede dar el paquete. Cada nodo que renvía el paquete debe reducir este valor en uno. Cuando el valor es igual a cero, el paquete se descarta.
Dirección de origen	128	Dirección IPv6 del nodo que originó el paquete.
Dirección destino	128	Dirección IPv6 del nodo destino final del paquete.

1.3. FORMATO DE UNA DIRECCIÓN IPv6

Una dirección IPv6 se compone de 8 campos de 16 bits cada uno, representado en formato hexadecimal; separados por ":" cada dos bytes.

Para simplificar la escritura de las direcciones IPv6, se utiliza un formato comprimido que sigue las siguientes reglas:

- Los ceros que preceden otros dígitos, en cada división de la dirección, pueden omitirse. Por ejemplo, la dirección "1234:ABC9:00C1:0001:0000:0000:0A0B:00FF" puede comprimirse de la siguiente forma "1234:ABC9:C1:1:0:0:A0B:FF".
- Una dirección IPv6 puede escribirse sin tantos ceros; cuando hay una sucesión de campos de ceros, esta puede remplazarse por la notación "::" , por ejemplo, la dirección "1234:ABC9:C1:1:0:0:A0B:FF" puede escribirse "1234:ABC9:C1:1::A0B:FF"[1]
- Las direcciones que consisten de solo ceros, por ejemplo 0:0:0:0:0:0:0:0, es una dirección no especificada, que puede ser escrita como "::". Esta dirección aparece en el enlace durante la autoconfiguración antes de la asignación de una dirección global.
- No hay diferencia entre mayúsculas y minúsculas, por consiguiente la dirección "1234:ABC9:C1:1::A0B:FF" equivale a "1234:abc9:c1:1::a0b:ff"

Para identificar las secciones de la dirección que identifican a la red y a los dispositivos, es utilizado un formato CIDR en la forma <dirección>/<prefijo>. Por ejemplo, una dirección de la forma 2001:abc:1234:cc9::1/64 indica que los primeros 64 bits identifican a la red (2001:abc:1234:cc9) y los restantes identifican a los dispositivos de dicha red (::1).

1.4. DIRECCIONAMIENTO

Se han definido tres tipos de direcciones en IPv6:

- *UNICAST:* Conexión punto a punto a través del camino más corto entre los terminales de origen y destino. Identifican a un nodo único y particular.
- *ANYCAST:* Comunicación punto a punto más cercano. Identifica a un grupo de nodos. El tráfico se envía al nodo más cercano al emisor.
- *MULTICAST:* Comunicación punto-multipunto. El tráfico enviado a una dirección "multicast" es enviado a todos los nodos pertenecientes al grupo.

Las direcciones de tipo broadcast, han sido remplazadas por las direcciones "multicast", ya que identifican a determinados grupos de dispositivos en una red.

[1] Esta regla puede usarse solo una vez dentro de una dirección IPv6, para que el sistema pueda determinar cuantos campos han sido comprimidos.

1.4.1. DIRECCIONAMIENTO UNICAST

Las direcciones unicast son direcciones IPv6 únicas globales en la red, identifican a cada nodo que se encuentre conectado a la misma. Se asocian a una interfaz en específico, por lo que no se encuentran duplicadas en la red. Por consiguiente, se consigue una conexión punto a punto entre los nodos.

A través de contextos se define el dominio de una red, ya sea lógico o físico; esta es una característica introducida en IPv6 para las este tipo de direcciones. Al reconocer el contexto al que pertenece una dirección se puede tener un manejo óptimo de los recursos de la red.

Las direcciones unicast se dividen en los siguientes contextos:

- Dirección local al enlace (link-local). Identifica a todos los nodos dentro de un enlace (capa 2). Cada interfaz debe tener una dirección link-local. No pueden ser enrutadas y solo son válidas al interior del enlace. Se obtiene automáticamente, sin necesidad de intervención del usuario.
- Dirección local única (unique-local). Identifica a todos los dispositivos dentro de una red interna o sitio, compuesta por varios enlaces o dominios capa 2. Equivalente a direcciones privadas en IPv4, es decir, proveen conectividad entre los nodos de un sitio o intranet. No pueden enrutarse hacia internet.
- Dirección de enlace global (global-link). Utilizada para distinguir cada host en la red. Pueden asignarse varias direcciones globales a una interfaz. Cada interfaz IPv6 tiene asignada tanto la dirección link-local como la dirección global. Usualmente son la misma dirección luego de la autoconfiguración. Comunican nodos a través de internet.

Tales contextos presentan una estructura jerárquica, siendo el contexto global el de mayor jerarquía, y el local el de menor.

Una interfaz IPv6 puede tener varias direcciones, una local al enlace para la comunicación con dispositivos locales, y una o más direcciones globales para comunicarse hacia internet.

1.4.2. DIRECCIONAMIENTO ANYCAST

Es una dirección que identifica un grupo de interfaces. Cuando una dirección unicast es asignada a varias interfaces, esta se convierte en una dirección anycast. Además debe configurarse en cada enrutador una ruta directa hacia tal dirección.

Los paquetes enviados a las direcciones anycast, se renvían a través de la infraestructura de enrutamiento hacia la interfaz más cercana al origen del paquete. Tal infraestructura, conoce las interfaces asociadas a las direcciones anycast y sus métricas de enrutamiento.

Estas direcciones solo son válidas como direcciones de destino en los paquetes IPv6.

1.4.3. DIRECCIONAMIENTO MULTICAST

El tráfico multicast opera de manera similar que en IPv4. Los dispositivos IPv6 ubicados en distintos lugares pueden recibir tráfico dirigido a una única dirección multicast.

La Figura 1.3 muestra la estructura que presenta una dirección multicast. El campo L indica el tiempo de vida de un grupo determinado, siendo 0 si es permanente, o uno cuando es temporal. El campo S indica el contexto o alcance de grupo, que puede ser interfaz, enlace, global, entre otros.

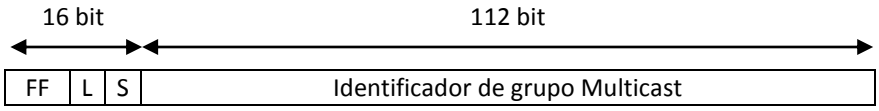

Figura 1.3. Estructura de dirección multicast

Algunas de las direcciones multicast son reservadas en IPv6, y se reciben durante la autoconfiguración del host. Con esto se consigue una selección más precisa de los destinatarios de una solicitud. Algunos de los grupos existentes se muestran en la Tabla 1.2.

Tabla 1.2. Direcciones de grupos multicast fijos

Dirección multicast	Descripción
FF01::1	Todos los nodos en la interfaz
FF02::1	Todos los nodos en el enlace
FF01::2	Todos los enrutadores en la interfaz.
FF02::2	Todos los enrutadores en el enlace
FF05::2	Todos los enrutadores en el sitio

1.5. ALGORITMOS DE ENRUTAMIENTO

Se han mantenido los modos de operación de los actuales protocolos de enrutamiento; pero se han desarrollado nuevas versiones que aprovechen las nuevas y mejoradas características de IPv6.

La Tabla 1.3 resume los protocolos de enrutamiento desarrollados para IPv6.

Tabla 1.3. Protocolos de enrutamiento para IPv6

Protocolo de enrutamiento	Versión IPv6
RIP	RIPng
EIGRP	EIGRP para IPv6
OSPF	OSPFv3
IS-IS	Integrated IS-IS
BGP	BGP-MP

2. CALIDAD DE SERVICIO

La calidad de servicio – *Quality of Service (QoS)* – se define como la capacidad que tiene una red para sostener un comportamiento adecuado del tráfico que transita por ella, proporcionando diversos niveles de servicio a los distintos tipos de tráfico. Con ella, se asegura la entrega de información, dando prioridad a las aplicaciones de desempeño crítico, como el tráfico de video y voz en tiempo real.

Permite el uso eficiente de recursos, durante congestiones de la red, seleccionando tráfico en específico, priorizándolo según importancia; y utilizando métodos de control y evasión de la congestión. QoS se basa en la clasificación o diferenciación de flujos de tráfico, y determina las formas en qué serán manejadas tales clases de tráfico a medida que circulan por la red.

Con la implementación de calidad de servicio en una red, se consigue un rendimiento mayor y más predecible, y aprovechamiento de ancho de banda existente.

2.1. ARQUITECTURA DE CALIDAD DE SERVICIO

Existen tres niveles bajo los cuales se trabaja la QoS dentro de una red: servicio al mejor esfuerzo, servicios integrados y servicios diferenciados. Los dos últimos, son arquitecturas estandarizadas por la IETF. Estos, proponen metodologías a través de las cuales se realiza la implementación de calidad de servicio, asimismo como dar garantía de su funcionamiento.

2.1.1. SERVICIO AL MEJOR ESFUERZO

Este es el tipo de servicio proporcionado por la red, al hacer todo lo posible para que un paquete alcance su destino, pero sin dar garantía de que ello suceda. De ser necesario, y sin solicitar permisos o dar notificación a la red, una aplicación enviará la cantidad de datos que desee. Este modelo es utilizado actualmente por aplicaciones FTB y HTTP.

2.1.2. SERVICIOS INTEGRADOS

Los servicios integrados ofrecen un nivel garantizado de servicio, a través de la reserva de recursos extremo a extremo.

La arquitectura de IntServ se basa en la idea de reservar de extremo a extremo, y por cada tipo de flujo, ancho de banda y recursos necesarios para que la aplicación pueda operar. Tales reservas se

mantienen hasta que la aplicación culmine, o se exceda del ancho de banda reservado para la misma. Esta fue la primera arquitectura propuesta para ofrecer calidad de servicio en IP.

IntServ es un protocolo que clasifica el flujo de tráfico dependiendo de la etiqueta de flujo establecida en la cabecera de un paquete IPv6. Los servicios de los diferentes flujos se clasifican en 3 tipos, para especificar el tratamiento que debe darse a cada flujo; tales tipos son: servicio garantizado, para tráfico en tiempo real; servicio de carga controlada, para tráfico en tiempo real menos crítico; y servicio al mejor esfuerzo.

El modelo está sustentado en los siguientes supuestos:

- Recursos gestionables directa y explícitamente para cumplir requerimientos de aplicaciones. Lo que conlleva al uso de mecanismos de control de admisión y reserva de recursos.
- Común infraestructura para tráfico regular y de tiempo real, que vendría siendo la Internet. Por lo que debe unificarse la pila de protocolos para todo tipo de tráfico.

Para cumplir con el modelo de reserva de recursos con antelación, IntServ se basa en el protocolo de Reservación de Recursos (RSVP), el cual, a través de un conjunto de mensajes de señalización realiza el transporte de información sobre los requerimientos y propiedades de cada flujo. Con esto, se mantienen las tablas de estado en cada uno de los nodos, generando alto tráfico de señalización y ocupación de recursos en los dispositivos.

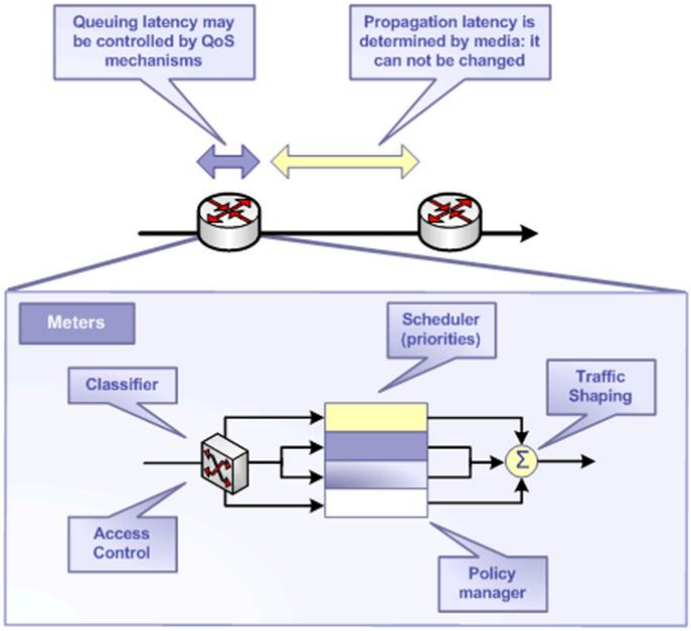

Figura 2.1. Estructura de IntServ

2.1.3. SERVICIOS DIFERENCIADOS

La arquitectura de servicios diferenciados (DiffServ), está basada en la idea que la prioridad relativa y sus marcas de tipo de servicio son suficientes para deducir el tratamiento de calidad de servicio que debe darse a los paquetes en cada enrutador.

El modelo de DiffServ busca dividir el tráfico en clases de servicio, dejando para los nodos de frontera del dominio, el procesamiento más complejo. Los requerimientos de QoS son especificados en un acuerdo de nivel de servicio (SLA), por lo que la reserva de recursos por flujo no es necesaria.

La manera como trabaja un nodo de DiffServ se representa como se aprecia en la Figura 2.2. Cuando un paquete IPv6 llega al nodo (enrutador), se clasifica en una de las clases identificadas por el mismo –este actuaría como clasificador–, estas clases puede ser por direcciones de red, protocolo, puertos, interfaz de ingreso, o cualquiera dada por el uso de listas de acceso. Luego de la clasificación, el paquete debe ser correctamente marcado; el marcador cambia la clase de prioridad. El medidor trata de definir parámetros de calidad como probabilidad de pérdida, retardo, y jitter; con esto estimar la calidad ofrecida a los usuarios. Si se da el caso en el cual el paquete se retarda más del requerido, o hay probabilidad de pérdidas más alta que la estipulada, el shaper (moldeador) tratara de corregir el tráfico, y por ende las características de calidad.

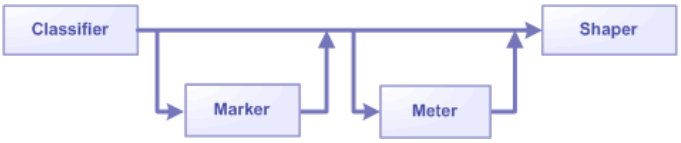

Figura 2.2. Configuración de DiffServ

DiffServ no es considerado un servicio de extremo a extremo; el tráfico es clasificado de modo que los enrutadores internos le asignen un comportamiento de renvío predeterminado, conocido como comportamiento por saltos (PHB – *Per Hop Behavior*); utilizando el campo de clase de tráfico del encabezado de un paquete IP, para diferenciar los distintos comportamientos. El campo del encabezado es cambiado por un código de 8 bit denominado DSCP – *DiffServ Code Point* –, en donde: 6 bits está destinados para diferenciar las clases de tráfico, y 2 bits reservados. Este código se asigna en los terminales, o en el enrutador de ingreso al dominio DiffServ, y es examinado por cada uno de los nodos sobre la ruta, para poder gestionar colas, y controlar mecanismos de clasificación en los enrutadores.

Se han definido dos PHB adicionales al mejor esfuerzo.

- *Renvío Expedito (EF)*. Son servicios de baja pérdida de paquetes, bajo retardo, bajo jitter y ancho de banda asegurado. Código de marcado: 101110

- *Renvío asegurado (AF)*. Son servicios con parámetros iguales al SLA, pero se permite mayor generación de tráfico que el establecido, donde el excedente no será tratado del mismo modo. Por esto se utilizan 4 clases de AF con marcas que dan a conocer el orden en que se eliminaran paquetes en caso de presentarse congestión.

Las ventajas obtenidas al utilizar DiffServ son principalmente, la operación más rápida de los enrutadores debido a la limitación en la complejidad de clasificación y encolado; además se minimiza el tráfico de señalización y almacenamiento. Finalmente se consigue mayor escalabilidad.

2.2. MÉTODOS DE CALIDAD DE SERVICIO

En una red IP, es posible proporcionar calidad de servicio gracias a ciertas metodologías implementadas, tales como las estrategias de manipulación de paquetes dado el caso se presenten congestiones; o evitar que se alcance tal estado, descartando paquetes en el momento que estos ingresan a la red.

Adicionalmente se implementan políticas de modelamiento, manipulación y clasificación de tráfico, para poder administrar eficientemente los recursos de la red.

2.2.1. ADMINISTRACIÓN DE LA CONGESTIÓN

La administración o manejo de la congestión, es un término utilizado para denotar las diferentes estrategias de encolamiento para el manejo de situaciones donde la demanda del ancho de banda solicitado por las aplicaciones, excede el ancho de banda total que puede proporcionar la red; llevando un control sobre el tráfico entrante a una red, estableciendo para ciertos flujos, prioridad sobre otros.[2]

Los tipos de encolamiento son:

- FIFO
- FAIR QUEUING
- encolamiento de prioridad
- Encolamiento personalizado
- Encolamiento de baja latencia
- MDDR
- CLASS-BASED-WEIGHTED-FAIR-QUEUING

[2] Álvarez Moraga, Sebastián A.; González Valenzuela, Agustín J. Título: "Estudio y configuración de calidad de servicio para protocolos IPv4 e IPv6 en una red de fibra óptica WDM".

2.2.1.1. FIFO

FIFO es el tipo de encolamiento más sencillo, el cual consiste en un buffer que retiene los paquetes entrantes, hasta que la interfaz de transmisión pueda enviarlos; basándose en el concepto de primer paquete entrante, primer paquete saliente. Maneja una cantidad limitada de flujos de datos por lo que al llegar paquetes cuando la cola está llena, estos se descartan. Adicionalmente, no tiene mecanismos de diferenciación de paquetes. El comportamiento de una cola FIFO es muy predecible. Los paquetes no son reordenados y el retardo máximo viene determinado por el tamaño máximo de la cola.

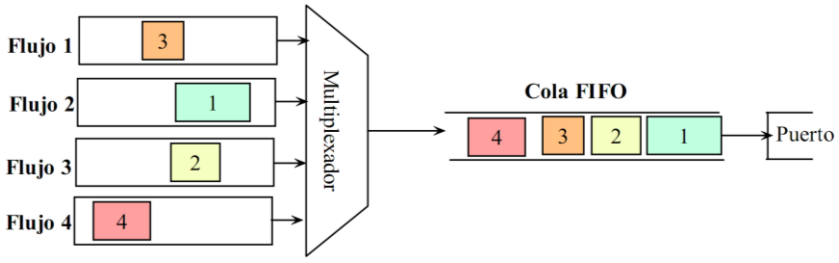

Figura 2.3. Funcionamiento de FIFO

2.2.1.2. FAIR-QUEUING (FQ)

Es un mecanismo que provee una justa asignación de ancho de banda para cada flujo de tráfico dentro de la red, de forma que determina el orden de tránsito en la cola de paquetes. La ponderación por flujos es realizada a través de filtros disponibles en TCP/IP, como dirección IP de origen y destino, tipo de protocolo, puerto TCP/UDP, o ToS de IP.

FQ crea una cola diferente para cada tipo de tráfico, y utiliza un valor determinado para la profundidad que debe tener la cola. Flujos de bajo volumen, sensibles al retardo, serán ubicados al inicio de la cola.

Las colas se sirven siguiendo un tiempo en orden round-robin, es decir, en orden secuencial circular (del primero al último y vuelta al primero). Las colas vacías se saltan. FQ se denomina también per-flow o flow-based queueing[3].

[3] Alarcón Llamas, Ricardo. Título: "Estudio e implementación de mecanismos de calidad de servicio sobre una arquitectura de servicios diferenciados". Universidad Politécnica de Cartagena. Enero 2003. Pág. 23.

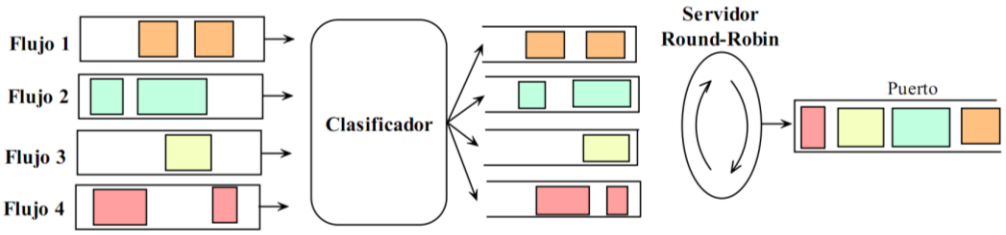

Figura 2.4. Funcionamiento de FQ

Es apropiado en situaciones donde se desea proveer un tiempo de respuesta consistente ante usuarios que generan altas y bajas cargas en la red. Debido a que cada flujo tiene una cola asignada, si este presenta demasiadas tramas de datos o intenta consumir más ancho de banda, solo se verá afectado el rendimiento de su cola debido a que son independientes del resto del tráfico.

Es una técnica poco escalable puesto que requiere recursos adicionales en la clasificación y manipulación dinámica de las colas. Además, FQ no está diseñado para soportar un número de flujos con diferentes requerimientos de ancho de banda. Es sensible al orden de llegada de los paquetes. Si un paquete llega a una cola vacía inmediatamente después de que la cola sea visitada por el servidor round-robin, el paquete tendrá que esperar en la cola hasta que todas las otras colas se sirvan antes de poder ser transmitido.

2.2.1.3. Encolamiento de prioridad (PQ)

Es una metodología a través de la cual se ofrece un tratamiento preferencial a paquetes, que en el momento de ingresar a la interfaz, son identificados por prioridad. Cada paquete se asigna a una de las colas disponibles, que son tratadas en estricto orden de prioridad.

Los paquetes se sirven de la cabecera de una cola, sólo, si todas las colas de prioridad mayor están vacías. Dentro de cada una de las colas de prioridad, los paquetes se sirven en el orden FIFO.

PQ se ajusta a condiciones donde existe tráfico importante, pero puede causar la total falta de atención de colas de menor prioridad.

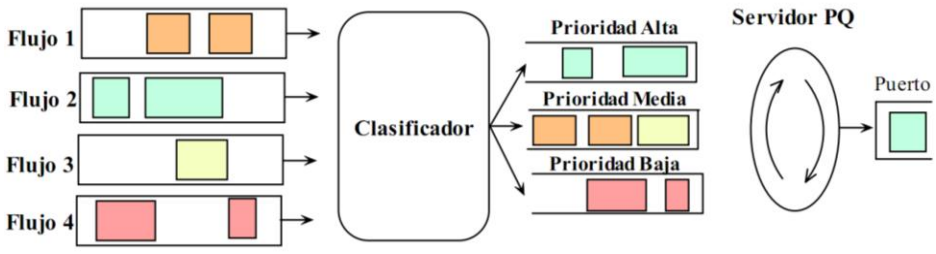

Figura 2.5. Funcionamiento de PQ

Permite a los enrutadores organizar los paquetes almacenados y por tanto servir una clase de tráfico de modo diferente a otras. Por ejemplo, se pueden colocar prioridades a las aplicaciones de tiempo real, como voz y video interactivo, y que se traten de forma prioritaria frente a otras aplicaciones que no operan en tiempo real. PQ no es la solución a las limitaciones del encolamiento FIFO en donde se favorecían a los flujos UDP sobre los TCP, durante periodos de congestión.[4]

2.2.1.4. Encolamiento Personalizado (CQ)

El encolamiento personalizado, o cola de prioridad, es un mecanismo establecido para priorizar el tráfico, evitando la inanición de las colas de menor prioridad, en donde se especifica el número de paquetes que deben atenderse por cada cola[5]. Cabe resaltar que no asegura prioridad absoluta como PQ.

Es empleado para proporcionar ancho de banda a tráficos en particular, en un punto de posible congestión. Asegurando una porción de ancho de banda al tráfico que lo amerite, y permitiendo el uso de los recursos disponibles al resto del tráfico.

2.2.1.5. Encolamiento de baja latencia (LLQ)

Es el método de encolamiento recomendado para voz sobre IP (VoIP) y telefonía IP. Consta de colas de prioridad personalizadas basadas en clases de tráfico, junto con una cola de prioridad que tiene preferencia sobre el resto de colas.

Debe configurarse ancho de banda límite reservado para la cola de prioridad. Esta cola da un máximo de retardo garantizado para los paquetes entrantes a la misma, que se calcula como el tamaño del MTU dividido por la velocidad del enlace.

2.2.1.6. MDRR

Al configurarse MDRR (*Modified-Deficit-Round-Robin*) para encolamiento, las colas que no están vacías se atienden una tras otra en forma de round robin. Cada vez que se atiende una cola, MDRR hace seguimiento de la cantidad de datos desencolados por encima del valor configurado.

Para compensar este excedente, al atender nuevamente la cola, son desencolados una menor cantidad de datos que en el turno anterior. Entonces, la cantidad promedio de datos atendidos por cola, será aproximadamente igual al valor configurado.

MDRR mantiene una cola prioritaria que es atendida de manera preferencial.

[4] Alarcón Llamas, Ricardo. OP. Cit. Pág. 22.

[5] Álvarez Moraga, Sebastián A.; González Valenzuela, Agustín J. OP. Cit. Pág. 4.

2.2.1.7. CLASS-BASED-WEIGHTED-FAIR-QUEUING (CBWFQ)

El tipo de encolamiento FQ colapsa debido a la cantidad numerosa de flujos que analiza, por lo que presenta ciertas limitaciones de escalamiento.

CBWFQ es una expansión del algoritmo, el cual permite al usuario crear clases con las que se consiga mayor control sobre las colas de tráfico y asignación de ancho de banda.

Cada clase tiene una cola separada, a la cual ingresan paquetes que cumplen con el criterio definido por cada clase. Para cada una pueden configurarse ancho de banda, límite de paquetes o profundidad de cola, entre otros.

2.2.2. EVASIÓN DE LA CONGESTIÓN

Las técnicas de evasión de la congestión monitorean constantemente el flujo de tráfico dentro la red con la finalidad de anticipar y minimizar su impacto sobre la misma. Están basadas en el modo que operan los protocolos para evitar alcanzar el nivel de congestión en la red.

Cuando operan múltiples conexiones TCP sobre un mismo enlace, estas incrementan el tamaño de su ventana deslizante a medida que llega el tráfico. Tal aumento conlleva a un mayor consumo de ancho de banda hasta el punto de crear congestión; es entonces, cuando las conexiones TCP presentan errores de transmisión, reduciendo nuevamente el tamaño de su ventana; generando el efecto conocido como *sincronización global*. El cual equivale a alcanzar el estado de congestión, mediante el incremento de la tasa de transmisión por cada flujo.[6]

Al presentarse estos períodos de congestión, entran en función las técnicas implementadas. Una de las técnicas más usadas es *tail drop,* que consiste en descartar paquetes entrantes a una interfaz, cuando la cola está llena; y continuarán siendo descartados hasta que haya espacio disponible en la cola, tal y como se encuentra representado en la Figura 2.6. Es la técnica más fácil de implementar, pero no se tiene control sobre los paquetes que se descartan, por lo cual pueden presentarse inconvenientes en el momento de recibir ráfagas de tráfico si la cola se encuentra llena, y no se podrá almacenar la información para transmitirla luego. Además, los usuarios no reconocen el estado de congestión tan solo con descarte de paquetes.

[6] Álvarez Moraga, Sebastián A.; González Valenzuela, Agustín J. OP. Cit. Pág. 4

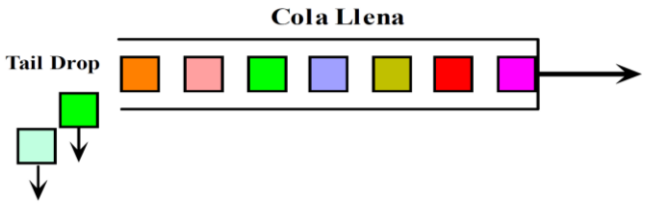

Figura 2.6. Funcionamiento de Tail Drop

Ahora bien, existen gestores de cola que permiten a los enrutadores responder activamente al incremento del tamaño de sus colas, marcando o descartando paquetes antes de consumir los recursos que tengan disponibles. Con esto se consigue eliminar el efecto de sincronización global de las fuentes TCP, y se tiene cierto control sobre el tamaño de las colas al influir en el retardo de encolamiento a través del enrutador.

El mecanismo de RED – *Random Early Detection*, monitorea constantemente el tamaño de la cola, de modo que cuando esta alcanza un umbral determinado, selecciona aleatoriamente paquetes entrantes para ser descartados, indicando así al emisor, que reduzca el tamaño de su ventana de transmisión para que el buffer del enrutador no se desborde.

Uno de los retos de la implementación de RED es la selección de la metodología para calcular la congestión, los cuales se diferencian en el cálculo del grado de ocupación de la cola.

El mecanismo de WRED – *Weighted Random Early Detection*, combina las capacidades del algoritmo de RED con la precedencia IP. Permite asignar diferentes perfiles de descarte a diferentes tipos de colas o tráfico.

Cuando un paquete ingresa a la interfaz se calcula el tamaño medio de la cola, que equivale a $Media = \left(Media_{anterior} \cdot \left(1 - \frac{1^n}{2}\right)\right) + \left(Tamaño_{actual} \cdot \frac{1^n}{2}\right)$, donde n es el factor de peso exponencial configurado por el usuario. Entonces: si la media es menor que el umbral mínimo del tamaño de la cola, el paquete es encolado; si está entre el umbral mínimo y máximo del tamaño de la cola, el paquete puede ser encolado o descartado dependiendo de la probabilidad de descarte del mismo; y si la media supera el umbral máximo de la cola, el paquete se descarta automáticamente.[7]

El limitante de estas técnicas de evasión de congestión es que están establecidas para tráficos TCP, puesto que otros protocolos no utilizan el concepto de ventana deslizante.

[7] Alarcón Llamas, Ricardo. OP. Cit. Pág. 32.

2.2.3. POLICING Y MODELAMIENTO DE TRÁFICO

Para administrar eficientemente los recursos de una red, se necesita de la limitación del tráfico saliente en una interfaz determinada. Esta limitación se realiza mediante las metodologías de Policing y modelamiento de tráfico.

Policing especifica la limitación a un máximo de tasa de transmisión o recepción para una clase de tráfico, controlando el ancho de banda del enlace. Es configurado sobre los extremos de la red. Cuando el tráfico entrante excede el umbral configurado, se descartan los paquetes, o se transmiten con una prioridad diferente. Como resultado se tiene que el tráfico total nunca excede el nivel predefinido, además no se pueden almacenar paquetes para enviarlos más adelante. La Figura 2.7 muestra cómo opera la técnica de Policing a lo largo del tiempo, cuando el tráfico excede lo predefinido.

Al controlar la tasa de salida con descarte de paquetes, se reduce el retardo por encolamientos. Debido a los descartes realizados, el tamaño de la ventana deslizante de TCP se reduce, ofreciendo rendimiento global de tráfico.

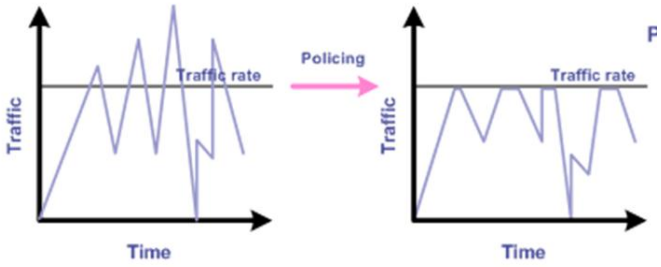

Figura 2.7. Modo operación de Policing

En cambio, las técnicas de modelamiento de tráfico, no descartan el tráfico excedente determinado por la tasa, sino que retrasan parte del mismo a través de colas, ajustándolo a un perfil determinado. Tiene un buffer finito, donde los paquetes se descartan cuando no hay espacio suficiente en el para almacenar los paquetes retardados.

Es una buena herramienta para cuando deba respetarse cierta tasa máxima de transmisión. Al adicionar retardos variables, se consigue un flujo de paquetes más sincronizado. No tiene en cuenta a usuarios finales para llevar a cabo sus funciones. La Figura 2.8 muestra cómo opera la técnica de Traffic shapping (TS) en el tiempo, se puede apreciar un flujo más suavizado cuando se alcanza el umbral de transmisión.

Es posible modelar tráficos de web o FTP a velocidades inferiores a las del receptor, puesto que los procedimientos realizados en Traffic Shapping, son independientes de la velocidad real del circuito.

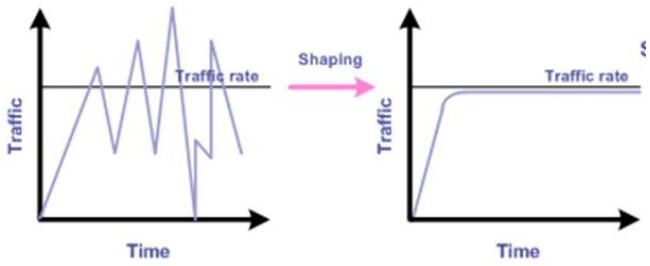

Figura 2.8. Modo operación Traffic Shapping

Ocasionalmente se necesita utilizar una vía con la velocidad adecuada para la transmisión de paquetes de alta o baja prioridad. A esta diferenciación se le conoce como enrutamiento basado en políticas (PBR – *Policy Based Routing*); y es implementado mediante listas de acceso donde se selecciona el tráfico crítico.

2.2.4. MANIPULACIÓN Y CLASIFICACIÓN DE TRÁFICO

La clasificación es utilizada para separar paquetes según ciertas características, predefiniendo patrones en el campo de ToS; como también en información de protocolos de nivel superior. El marcado se conoce como la acción en la cual el campo de ToS puede remplazarse por un valor relevante a las políticas de QoS definidas en la red.[8]

Ahora, para poder manipular los tráficos y ofrecerles calidad de servicio, se utilizan procedimientos básicos de clasificación y asignación de prioridad, denominados Mapas de clase y Mapas de política.

- *Mapas de clase:* mecanismo usado para nombrar y aislar un flujo de tráfico específico, que define el criterio utilizado para comparar el tráfico y luego clasificarlo.
- *Mapa de política:* específica en qué clase de tráfico actuará. Las acciones que puede tomar son confiar en valores de clase de servicio, DSCP o precedencia IP de la clase de tráfico; establecer un valor específico de estos o especificar límites de ancho de banda. Antes que un mapa de política sea efectivo, debe pertenecer a una interfaz.

2.3. CALIDAD DE SERVICIO EN IPv6

Actualmente la calidad de servicio de una red es implementada a través del campo ToS, cuya función es especificar parámetros de prioridad, retardo, rendimiento y fiabilidad, requeridos para

[8] Salcedo Parra, Octavio J.; López, Danilo; Ríos, Ángela. Título: "Desempeño de la calidad de servicio sobre IPv6". Artículo de investigación Conciencias. Febrero 1 de 2011. Pág. 5.

un nivel de servicio dado. De este modo los paquetes con diversas opciones de ToS, pueden manejarse con diferentes niveles de servicio dentro de la red.

Estos parámetros son usados para especificar el tratamiento del datagrama durante su transmisión en una red.

El campo de ToS se compone por un campo de precedencia, que es una medida de importancia relativa al datagrama, dando tratamiento preferencial a los que tengan precedencia superior; tres indicadores D, T, R, que especifican lo que realmente interesa entre el retardo, rendimiento o fiabilidad; y dos bits no utilizados.

El uso de indicadores puede incrementar el coste del servicio. En muchas redes el mejor desempeño de uno, significa el peor desempeño del otro; por lo que no se deben establecer más de dos indicadores prioritarios.

Aun cuando el campo ToS de IPv4 se utilizó para el marcado de paquetes con un nivel de servicio requerido, se presentaron ciertos inconvenientes o ambigüedades por su significado, por lo que se cambió al campo DSCP (*Differentiated Services Code Point*). Tal fue el éxito de esta definición que se incluyó para ofrecer los mismos beneficios en IPv6, a través del campo de clase de tráfico – TC[9]; mostrado en la

Figura 2.9. Además se tiene el campo de etiqueta de flujo, como herramienta para implementar QoS en IPv6.

El campo de etiqueta de flujo (20 bits), es agregado para permitir el marcado de paquetes pertenecientes a flujos de tráficos particulares, o para marcar secuencias de paquetes que solicitan manejo especial por parte de los enrutadores IPv6.

El campo de clase de tráfico se utiliza para identificar y distinguir entre las diferentes clases o prioridades de paquetes, y tiene los siguientes requisitos:

- El valor de los 8 bits de clase de tráfico, de los paquetes asignados por nodo, es dado por un protocolo de capa superior, a través de un medio proporcionado por la interfaz de servicios para IPv6.
- Si un nodo tiene soporte de alguno de los bits de TC, es libre de modificarlos en los paquetes que envíen, renvíen o reciban, dependiendo del uso que requieran.
- Los protocolos de capa superior deben siempre revisar el valor de los bits de TC; nunca asumiendo el mismo valor de los bits del paquete recibido, y del enviado por el origen.

[9] TC – Traffic Class. Campo de clase de tráfico.

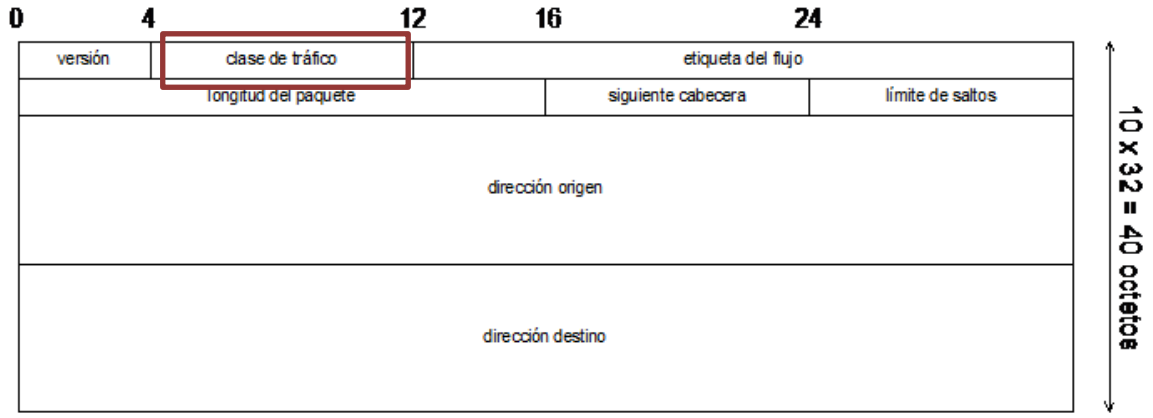

Figura 2.9. Cabecera IPv6

2.4. PARÁMETROS QUE MIDEN CALIDAD DE SERVICIO PERCIBIDA

Los elementos que hacen parte de la red, como los nodos y los enlaces entre los mismos, tienen cierto impacto sobre la calidad de servicio que perciben los usuarios en su propia red. Por consiguiente debe de analizarse la influencia que estos elementos tienen sobre los atributos de la QoS, tales como: el caudal (*Throughput*); retardos, varianza de retardo, y pérdidas de paquetes.

2.4.1. CAUDAL O THROUGHPUT

Es la medida de volumen de trabajo o de información que fluye a través de un sistema. Describe la capacidad que tiene un sistema para la transferencia de datos. Es sinónimo de consumo de ancho de banda digital.

Dentro de las redes TCP/IP, el Throughput se mide por la tasa de bytes o paquetes (libres de errores):

- Que fluyen por el circuito.
- De una aplicación específica.
- Del conjunto de flujos de un nodo a otro.
- Del conjunto de flujos de una red a otra.

Por deducción, entre mayor sea el valor de Throughput medido, mejor QoS es ofrecida, puesto que la tasa de transmisión de paquetes sin errores, es mayor.

El parámetro que un enrutador puede configurar para controlar el caudal, es la cantidad de ancho de banda reservado para los diferentes tipos de paquetes.[10]

Por ejemplo, un enrutador no controla la cantidad de ancho de banda asignado para el servicio de *Best Effort*. Por tanto, cuando se presentan congestiones los paquetes son colocados en una cola FIFO. Pero los datagramas UDP no reducen su tasa de transmisión, quedándose así con el ancho de banda total del enlace; aumentando el Throughput percibido por sus fuentes, y reduciendo el de las fuentes TCP.

Al diferenciar el tráfico en clases, creando colas por cada una, se controla el ancho de banda reservado por tráfico. Entonces, en casos de congestión, los flujos de tráfico UDP no utilizaran todo el ancho de banda del enlace, y podrá repartirse el caudal entre los flujos de tráfico.

2.4.2. RETARDOS (DELAY)

El retardo es la cantidad de tiempo que se toma transmitir un paquete de un punto de la red a otro, que es afectado por factores como enrutamiento, encolamiento, propagación y serialización. Es una medida que identifica el camino físico más que al nivel de congestión de la red.

Es necesario mantener métricas del retardo por varias razones. Por ejemplo, existen aplicaciones que no presentan buen funcionamiento si se tiene un valor alto de retardo en relación a un umbral determinado; aplicaciones en tiempo real con variaciones irregulares en retardo no son manejables; entre mayor sea el retardo, es más complicado para los protocolos de capa superior sostener altos anchos de banda; valores de esta métrica por encima del mínimo provee una indicación de la congestión presente en el camino; entre otras razones más.

Los parámetros que suelen medirse con relación al retardo son OWD y RTT.

2.4.2.1. OWD – ONE WAY DELAY

Es el tiempo que le toma a un paquete alcanzar su destino. Es considerado propiedad del enlace o el camino de la red.[11] Puede descomponerse en retardo por salto en una dirección, y estos pueden a su vez convertirse en componentes de retardo por nodo y por enlace. OWD equivale a la suma de los retardos individuales por los que atraviesa el paquete transmitido (retardo de extremo a extremo). La Figura 2. muestra una representación de lo mencionado, además se puede apreciar que los retardos pueden estar asociados tanto al nodo como al enlace.

[10] Alarcón Llamas, Ricardo. OP. Cit. Pág. 6.

[11] Leinen, Simon. One way delay. http://kb.pert.geant.net/PERTKB/OneWayDelay. Consulta: Diciembre 10, 2011.

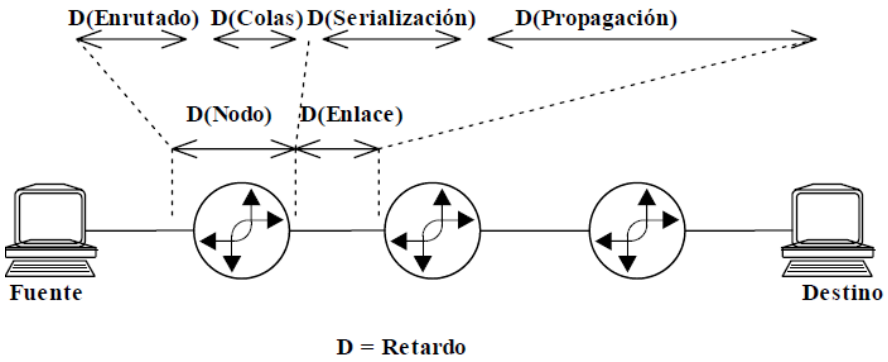

Figura 2.9. Retardo de extremo a extremo

- ***Retardo de enrutamiento.*** Asociado al nodo. Es la cantidad de tiempo que tarda un enrutador en recibir un paquete, tomar una decisión de encaminamiento y transmitir el paquete a través de un puerto de salida no congestionado. Medido en decenas o cientos de microsegundos.
- ***Retardo en colas.*** Asociado al nodo. Tiempo que espera un paquete en cola. Este retardo puede controlarse a través de gestión de memorias y servicios de colas.
- ***Retardo de propagación.*** Asociado al enlace. Tiempo para atravesar enlace físico. Medido en milisegundos. Sólo puede controlarse a través de la distancia de un enlace.
- ***Retardo de serialización.*** Asociado al enlace. Tiempo para colocar bits de un paquete en el cable cuando el enrutador lo transmite. Medido en milisegundos, en función del tamaño del paquete (en bits) y la velocidad del puerto (bps). Es controlado con interfaces de alta velocidad del enrutador.

La descripción de cualquier método de medida debe incluir un análisis de las fuentes de error e incertidumbre, entre las que se encuentran: sincronización entre relojes de fuente y destino; y la incertidumbre agregada por la resolución de cada reloj.

2.4.2.2. RTT – ROUND TRIP TIME DELAY

Es el tiempo total que le toma a un paquete enviado desde un nodo A, alcanzar su destino B, y enviar respuesta desde el nodo B hasta el origen A. Es la suma de los OWD de A-B y de B-A, y del tiempo que le toma a B enviar la respuesta.

Presenta facilidad de despliegue, ya que es posible realizar medidas de RTT sin instalar software para medir en el destino, con aproximaciones como ICMP o metodologías basadas en TCP.

También tiene facilidad de interpretación. No es muy preciso deducir RTT a partir de OWD, ya que muchas aplicaciones utilizan RTT para aproximar distancia y estimar localización de los hosts en internet.

Esta medida de retardo es importante para conexiones en la capa de transporte. En TCP es necesario conocer el tiempo de asentimiento de datos antes de retransmitir.

2.4.3. VARIACIÓN DEL RETARDO (JITTER)

Jitter, es la variación del retardo en el tiempo entre paquetes consecutivos que forman parte del mismo flujo. Puede medirse a través de ciertas técnicas, incluyendo la media, la desviación típica, máximo o mínimo tiempo de llegada entre paquetes.

La principal fuente de jitter es en las colas para paquetes consecutivos de un mismo flujo. Otra fuente potencial es que los paquetes consecutivos de un mismo flujo sigan caminos físicos diferentes. Además el jitter crece exponencialmente con el aumento de la utilización del ancho de banda al igual que el retardo. Por consiguiente el jitter influye en la calidad de servicio percibida, sobretodo para aplicaciones de voz o vídeo.

3. DISEÑO DE UN ESQUEMA DE CALIDAD DE SERVICIO PARA IP

Para poder brindar calidad de servicio en una red, deben tomarse en consideración los acuerdos de nivel de servicio establecidos (SLA) y los requerimientos de tráfico solicitados dentro de la red. Un esquema de calidad de servicio, se diseña con el propósito de administrar los enlaces, de tal modo que se asegure el cumplimiento de ancho de banda contratado, para evitar períodos de congestión que incrementen la tasa de paquetes descartados (o perdidos), disminuyendo la eficiencia de la red.

A través de los procedimientos de administración de ancho de banda en los enlaces, se elimina el tráfico que quede por fuera del perfil contratado. No obstante, si un paquete marcado con prioridad alta queda por fuera del perfil contratado, no se descartará, sino que se encolarán durante periodos de congestión.

Al simular el esquema diseñado de calidad de servicio, serán seleccionados los parámetros de configuración de los procedimientos para ofrecer las diferentes categorías de tráfico, garantizando con esto, el cumplimiento de las clases de servicio establecidas.

3.1. ACUERDO DE NIVEL DE SERVICIO

Para el modelamiento del esquema, se establecerán cuatro clases de servicio (distintas al de mejor esfuerzo), conforme a parámetros como tasa de paquetes entregados, latencia y jitter. Las categorías ofrecidas serán: Premium, Oro, Plata y Bronce. La Tabla 3.1 muestra los parámetros comprometidos para cada clase de servicio.

Tabla 3.1. Parámetros por clase de servicio

Parámetro	Clase de servicio			
	Premium	Oro	Plata	Bronce
Paquetes entregados	99,90%	99,50%	99,00%	-
RTT	≤ 150 ms	≤ 150 ms	-	-
Jitter	≤ 30 ms	-	-	-

3.2. ELECCIÓN DE LA ARQUITECTURA DE QoS

Como se mencionó en el capítulo anterior, sección 2.1, de acuerdo a la IEFT se tienen dos arquitecturas de servicios que permiten establecer QoS en equipamientos de red, con diferentes modos de operación: servicios integrados y servicios diferenciados. Ambas soportan el protocolo IP.

La arquitectura de servicios diferenciados ofrece ciertas ventajas sobre la arquitectura de servicios integrados, tales como su buen funcionamiento, flexibilidad, escalabilidad, entre otras. El tratamiento diferenciado de los agregados de tráfico es la filosofía sobre la que se basa esta arquitectura para brindar QoS a las redes IP.

Por consiguiente, se selecciona la arquitectura de DiffServ como la base para el desarrollo del esquema de QoS que será implementado en el documento.

3.3. PHB

En la sección 2.1.3 se expuso la metodología utilizada por DiffServ para ofrecer QoS, a través de un comportamiento de renvío predeterminado, o comportamiento por saltos (PHB). Con ellos se define el encolamiento y el tratamiento de retransmisión que recibirá un paquete perteneciente a un grupo de tráfico.

Existen cuatro estándares disponibles de PHBs con sus respectivos valores DSCP. Se utilizarán sólo tres estándares con algunos valores DSCP escogidos para clasificar el tráfico de la red en el diseño, de acuerdo a las categorías y parámetros indicados en la sección 3.1.

3.3.1. EXPEDITED FORWARDING (EF) PHB

Definido en el RFC 2598, tiene un valor DSCP igual a 101110, ofreciendo un servicio de bajas pérdidas, baja latencia, bajo jitter y ancho de banda asegurado para aplicaciones en tiempo real.

Para controlar la congestión utilizará tipo de encolamiento de LLQ, usando la cola de prioridad que tiene preferencia absoluta sobre las colas del mecanismo CBWFQ configurado en el AF PHB, atendiendo de manera inmediata el tráfico asignado.

No necesita mecanismo de evasión de congestión, puesto que tiene prioridad sobre cualquier otro tráfico. Al no configurar este mecanismo, se utiliza por defecto *tail drop*.

Corresponderá al servicio PREMIUM, ya que es el servicio con mejores prestaciones de calidad de servicio.

3.3.2. ASSURED FORWARDING (AF) PHB

Descrito en el RFC 2597 para asegurar que el tráfico conforme a un perfil establecido, se entregue sin pérdidas, definiendo 4 clases para reservar recursos y 3 categorías de descarte.

Se utilizarán las clases 1, 2 y 3, para los servicios Oro, Plata y Bronce, respectivamente; con dos categorías de preferencia de descarte, alto y bajo. La Tabla 3.2 contiene los valores DSCP asignados a cada clase de servicio definida previamente.

Tabla 3.2. Valores DSCP para cada CoS

% Descarte	Oro	Plata	Bronce
Bajo	AF11 = 001010 (10)	AF21 = 010010 (18)	AF31 = 011010 (26)
Alto	AF13 = 001110 (14)	AF23 = 010110 (22)	AF33 = 011110 (30)

Se utilizará tipo de encolamiento LLQ, utilizando colas de prioridad personalizadas con CBWFQ. Para especificar la cantidad de ancho de banda del enlace para cada una de las clases de servicio. Como mecanismo de evasión de congestión se configurará WRED.

3.3.3. PHB POR DEFECTO

Descrito en el RFC 2474, con un valor DSCP igual a 000000. Equivale al servicio del mejor esfuerzo tradicional. El tráfico de este PHB no recibirá ningún tipo de tratamiento especial.

3.4. TRÁFICO DE LA RED Y ASIGNACIÓN DE VALORES DSCP

Los diferentes tipos de tráfico serán clasificados conforme a los requerimientos que cada uno tiene para calidad de servicio. Por ende:

Dentro del servicio Premium, entran las aplicaciones con mayores exigencias de QoS, como VoIP y videoconferencia, que es la aplicación con un comportamiento bastante regular. Se establecerá un ancho de banda máximo de 800 kbps para períodos de congestión.

Dentro del servicio Oro, entraran aplicaciones de *streaming*, ya que los AF establecidos para este servicio cumplen con los requisitos de este tipo de tráfico (jitter, retardo y tasa de paquetes perdidos). Debe tener in 25 % del ancho de banda disponible en períodos de congestión.

Al servicio plata se le asignarán aplicaciones como transacciones, bases de datos, entre otras. Trabajará con un 20 % del ancho de banda.

En el servicio bronce estarán los protocolos y aplicaciones de tráfico para la administración de la red. Le será asignado un 10% del ancho de banda.

El servicio al mejor esfuerzo comprende el resto de aplicaciones, que no requieren tratamiento QoS.

Con lo mencionado, se pueden asignar los PHB a cada tipo de tráfico. Tal clasificación se muestra en la Tabla 3.3.

Tabla 3.3. Clasificación y asignación DSCP por tráfico

CoS	DSCP	Aplicación	%Buffer
Premium	EF = 46	VoIP, Videoconferencia	800 kbps
Oro	AF13 = 14	Streaming	25%
Plata	AF21 = 18	Bases de datos, SSL	20%
	AF23 = 22	HTTPS, SFTP	
Bronce	AF31 = 26	TELNET, SSH	10%
	AF33 = 30	DHCP, ICMP	
Mejor esfuerzo	BE = 0	-	

4. CONFIGURACIÓN Y SIMULACIÓN DEL ESQUEMA DE CALIDAD DE SERVICIO

En los servicios diferenciados se distinguen dos tipos de enrutadores: internos y de frontera. Los enrutadores de frontera se encargan de la clasificación y el marcado de tráfico, mientras que los enrutadores internos se encargan de evitar la congestión.

Para aplicar el esquema de calidad de servicio diseñado en el capítulo 3, se utilizarán dos enrutadores; uno que realice la diferenciación de servicios (actuando como nodo frontera); y el otro realizará una interconexión entre el primer router y otros hosts; creando así un cuello de botella en el enlace entre ambos enrutadores que provoque la congestión y permita apreciar como actuan las herramientas de evasión de la congestión.

4.1. ENTORNO DE PRUEBAS

Se implementará una estrategia de generación de tráfico desde las LAN hacia los enrutadores. Se utilizarán mecanismos de diferenciación de tráfico de cada una de las redes conectadas a las interfaces de los enrutadores, para que sean clasificados dentro de los grupos establecidos en el diseño del esquema de QoS.

Como se mencionó anteriormente, en el cuello de botella creado entre los enrutadores, se evaluaran las herramientas de manejo de congestión utilizadas por DiffServ.

El acondicionamiento de los flujos de tráfico entrante a las interfaces de los enrutadores, se marcarán los paquetes con DSCP dependiendo si cumplen o no con las características de flujo contratado.

Para evaluar las políticas de calidad de servicio que sean configuradas, se empleará una herramienta propia de Cisco, denominada IP SLA. Esta utiliza monitoreo de tráfico activo para monitorizar el tráfico a través de la red. Algunas de las características que pueden analizarse con IP SLA son: Jitter UDP, Jitter UDP para VoIP, conexión TCP, HTTP, ICMP, entre otras.

La Figura 4.1 resume el entorno sobre el cual se correrán las pruebas y las configuraciones.

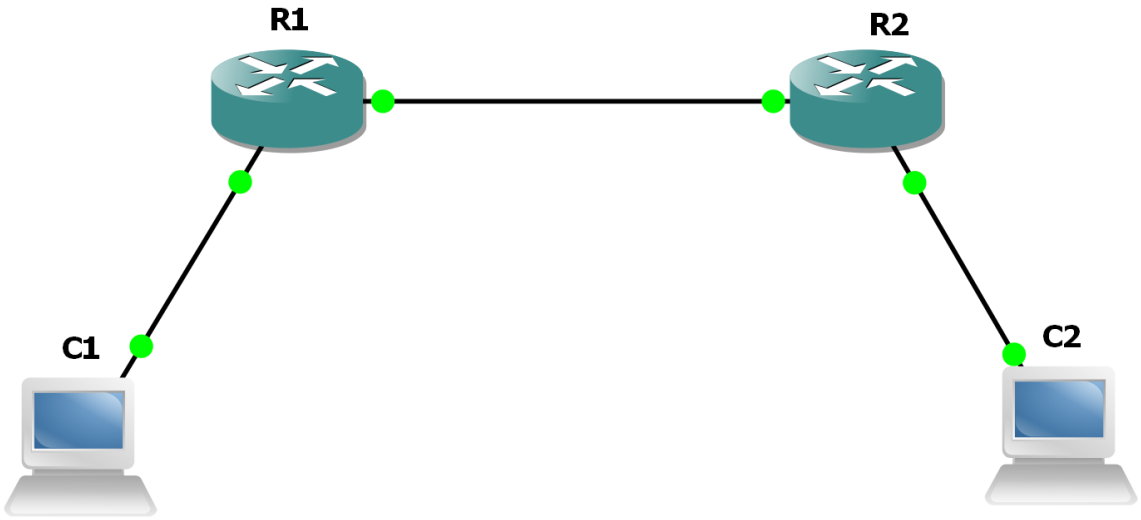

Figura 4.1. Esquema de pruebas

4.2. GNS3

GNS3[12] es un simulador de redes gráfico que permite la simulación de redes complejas. Está relacionado con Dynamips, emulador de IOS de Cisco; Dynagen; Qemu, emulador de maquinas virtuales; y VirtualBox, software de virtualización.

Es una aplicación realizada en Python, que utiliza las librerías de Dynagen para crear una interfaz gráfica (GUI). Sus principales funciones son realizar operaciones de la interfaz de línea de comandos (CLI).

Los puntos más destacados acerca de este simulador son:

- Es un software de libre distribución. De fácil instalación.
- Permite conexión Telnet a la consola de un router virtual, desde la interfaz gráfica.
- Permite comunicación entre redes virtuales con redes reales.
- Puede capturar los paquetes que pasan por enlaces virtuales y escribir los resultados de la captura de los archivos para que sean interpretados por aplicaciones como Wireshark o tcpdumps.

4.2.1. SIMULACIÓN DE HOSTS

Para la incorporación de equipos de red a las topologías creadas, GNS3 utiliza varias metodologías, como por ejemplo: utilizar Virtual PC; utilizar las interfaces de red disponibles en el equipo donde

[12] Graphical Network Simulator GNS3. Http://www.gns3.net/

se estén corriendo las pruebas; utilizando máquinas virtuales, e incluso utilizando un enrutador como host (limitando características de enrutamiento).

La herramienta utilizada será el programa Virtual PC (VPC), la cual utiliza puertos UDP para la comunicación entre el simulador y cada uno de los PC simulados. Es un software de distribución libre. Pero tiene funcionalidad limitada, puesto que solo permite el uso de comandos como "ping" y "traceroute".

Por lo tanto, para poder generar otros tipos de tráfico que requieran de tratamiento especial de QoS se utilizará una herramienta de Cisco denominada IP SLA, con la que, los enrutadores prueban los niveles de acuerdo de servicio configurados. Generando ciertas operaciones que arrojan los resultados correspondientes a los parámetros de QoS que son primordiales en cada caso.

4.2.1.1. Generación de Tráfico – IP SLA

Serán configuradas 9 operaciones de la siguiente manera:

- IP SLA 1 – UDP Jitter
- IP SLA 2 – UDP Jitter, conversación con codec g711ulaw
- IP SLA 3 – UDP Jitter, conversación con codec g729a
- IP SLA 4 – UDP Echo
- IP SLA 5 – Conexión TCP, Telnet
- IP SLA 6 – Conexión TCP, FTP
- IP SLA 7 – Conexión TCP, HTTPS
- IP SLA 8 – Conexión TCP, SSH
- IP SLA 9 – ICMP Echo

La operación **UDP Jitter**, fue diseñada para diagnosticar si una red es apta para aplicaciones de tráfico en tiempo real como VoIP, video sobre IP, o videoconferencia. Esta monitorea los cambios en jitter. Los paquetes que genera IP SLA tienen información de los paquetes de la secuencia enviada y la recibida. De acuerdo a esto, estas operaciones pueden medir: jitter en una dirección (de origen a destino y de destino a origen); pérdida de paquees; retardo en una dirección, y RTT promedio.

Esta operación funciona al generar tráfico UDP simulado (o sintético). Envía N paquetes UDP, cada uno de tamaño S, cada T milisegundos. Desde un enrutador fuente, a un enrutador destino. Por defecto envía tramas de 10 paquetes, cada 10 ms, y la operación se repite cada 60 segundos. Estas características pueden modificarse por el usuario.

Esta misma operación puede usarse para tráfico VoIP, ya que genera puntos aproximados de VoIP. Pero no proporciona soporte para RTP (Real-Time Transport Protocol – Protocolo de transporte en tiempo real). Se dan valores de ICPIF y MOS.

ICPIF – Calculated Planning Impairment Factor (Factor de degradación calculado). Cuantifica el factor de degradación para la calidad de voz encontrada en la red.

MOS – Mean Opinion Score. La calidad de la transmisión de voz es una respuesta subjetiva del que escucha. Este es una medida comúnmente usada para determinar la calidad del sonido producida por ciertos codecs.

Para obtener los valores de ICPIF y MOS deben configurarse codecs en la operación UDP Jitter. Los disponibles y sus valores por defecto se muestran en la siguiente figura. Esto serán los configurados para el IP SLA 2 y 3.

Codec	Default Request Size (Packet Payload) (s)	Default Interval Between Packets (t)	Default Number of Packets (n)	Frequency of Probe Operations (f)
G.711 mu-Law (g711ulaw)	160 + 12 RTP bytes	20 ms	1000	Once every 1 minute
G.711 A-Law (g711alaw)	160 + 12 RTP bytes	20 ms	1000	Once every 1 minute
G.729A (g729a)	20 + 12 RTP bytes	20 ms	1000	Once every 1 minute

Figura 4.2. Valores por defecto de la operación UDP Jitter por Codec

La operación **UDP Echo**, mide tiempo de respuesta de extremo a extremo entre el enrutador cisco y los dispositivos utilizando IP. Es útil para probar conectividad tanto para dispositivos Cisco, como para los que no son de esta marca.

El tiempo de respuesta RTT se computa al medir el tiempo entre el envío de un mensaje UDP Echo, de un router origen al destino, y la recepción de la respuesta al UDP echo del enrutador destino al origen.

La operación **TCP connect**, mide el tiempo de respuesta tomado para llevar a cabo una operación de conexión TCP entre el enrutador y los dispositivos de la red. Se configurarán operaciones de conexión TCP a Telnet, SSH, HTTPS y FTP.

Finalmente, la operación **ICMP Echo** mide el tiempo de respuesta extremo a extremo entre un enrutador Cisco y cualquier dispositivo usando IP. Este tiempo es computado al medir el tiempo entre el envío de un mensaje ICMP al destino, y la recepción de la respuesta ICMP Echo. Corresponde a los tiempos que se obtienen con la prueba ping ICMP.

4.2.2. EQUIPOS UTILIZADOS

Los enrutadores seleccionados son de la serie Cisco c7200, modelo c7200. Con una versión cisco IOS 15.0 (c7200-adventerprisek9-mz.150-1.M), que tiene soporte tanto para configuraciones IPv6, como para calidad de servicio.

Se le serán agregados en 1 slot una interfaz gigabit Ethernet, para conectarla con el otro enrutador; y 1 slot con 4 puertos Ethernet disponibles, donde serán conectados los PC.

4.3. CONFIGURACIÓN DE ESQUEMA DE QoS PARA IPv4

Luego de haber analizado las características de los equipos a implementar en la red, se procede con la descripción de los comandos de configuración para el desarrollo del esquema de calidad de servicio que fe diseñado.

Cisco ofrece ciertas herramientas para llevar a cabo la configuración de los comandos para el soporte de QoS. Una de esas posibilidades, que es la que será mayormente utilizada es MQC, configuración por modular QoS CLI, que permite la configuración de la calidad de servicio a partir de clases y políticas.

El esquema de pruebas general se muestra en la Figura 4.1, las direcciones asignadas a las interfaces de los enrutadores se muestran en la Tabla 4.1, y las direcciones asignadas a los hosts, con su respectiva puerta de enlace predeterminada, se muestran en la Tabla 4.2.

Tabla 4.1. Direcciones de las interfaces

	Interfaz	Dirección IP	Máscara	Ancho de Banda
Router 1	Gigabit Ethernet 0/0	10.10.12.1	24	2048 kbps
	Ethernet 1/0	192.168.12.1	24	256 kbps
Router 2	Gigabit Ethernet 0/0	10.10.12.2	24	2048 kbps
	Ethernet 1/0	200.172.16.1	24	256 kbps

Tabla 4.2. Direcciones IP de hosts

		Dirección IP
LAN 1	**PC 1**	192.168.12.7
	Máscara	255.255.255.0
	Gateway	192.168.12.1
LAN 2	**PC 2**	200.172.16.2
	Máscara	255.255.255.0
	Gateway	200.172.16.1

Se inicia introduciendo la información general en cada enrutador.

```
R1> enable
R1# configure terminal
R1 (config)# interface ethernet 1/0
R1 (config-if)# ip address 192.168.12.1 255.255.255.0
R1 (config-if)# bandwidth 256
R1 (config-if)# no shutdown
R1 (config-if)# exit
R1 (config)# interface gigabitEthernet 0/0
R1 (config-if)# ip address 10.10.12.1 255.255.255.0
R1 (config-if)# bandwidth 2048
R1 (config-if)# no shutdown
R1 (config-if)# exit
```

Como protocolo de enrutamiento se utilizará RIP, la configuración es la siguiente:

```
R1 (config)# router rip
R1 (config-router)# network 192.168.12.0
R1 (config-router)# network 10.10.12.0
R1 (config-router)# network 200.172.16.0
R1 (config-router)# end
R1# show ip protocol
R1# show ip route
```

Los últimos dos comandos son para la verificación de la configuración realizada. De manera análoga, se realiza la configuración del enrutador 2. Luego de esto se hace ping a todas las redes confirmando conectividad. Esto debe estar correcto antes de proceder con las configuraciones de QoS.

4.3.1. CREACIÓN DE LISTAS DE CONTROL DE ACCESO

A través de estas listas, será filtrado parte del tráfico que llega a las interfaces del enrutador para poder clasificarlo más adelante.

La configuración realizada para la creación de las mismas fue la siguiente:

```
R1 (config)# access-list 101 permit udp any any range 16384 32768
R1 (config)# access-list 102 permit tcp any any range 21 24
R1 (config)# access-list 103 permit tcp any any eq tacacs
R1 (config)# access-list 104 permit tcp any any eq www
```

```
R1 (config)# access-list 105 permit ip any any
R1 (config)# access-list 108 permit tcp any any eq telnet
R1 (config)# access-list 109 permit tcp any any eq 443
R1 (config)# access-list 110 permit icmp any any
```

4.3.2. CREACIÓN DE MAPAS DE CLASE

Se procede con la creación de las clases establecidas en la Tabla 3.3.

```
R1 (config)# class-map match-all Premium
R1 (config-cmap)# match ip dscp 46
R1 (config-cmap)# match access-group 101
R1 (config-cmap)# exit
R1 (config)# class-map match-all Oro
R1 (config-cmap)# match ip dscp 10 14
R1 (config-cmap)# match protocol rtsp[13]
R1 (config-cmap)# match protocol rtp[14]
R1 (config-cmap)# exit
R1 (config)# class-map match-all Plata
R1 (config-cmap)# match access-group 109
R1 (config-cmap)# match ip dscp 18 22
R1 (config-cmap)# match protocol secure-http
R1 (config-cmap)# match protocol secure-ftp
R1 (config-cmap)# match protocol ipsec
R1 (config-cmap)# exit
R1 (config)# class-map match-all Bronce
R1 (config-cmap)# match ip dscp 26 30
R1 (config-cmap)# match protocol dhcp
R1 (config-cmap)# match protocol icmp
R1 (config-cmap)# match protocol snmp
R1 (config-cmap)# match access-group 108
R1 (config-cmap)# match access-group 110
R1 (config-cmap)# exit
R1 (config)# class-map match-all best-effort
R1 (config-cmap)# match access-group 105
R1 (config-cmap)# exit
```

Para el caso del enrutador 2, se realiza la siguiente configuración:

```
R2 (config)# class-map match-all Premium
R2 (config-cmap)# match ip dscp 46
```

[13] RTSP: *Real Time Streaming Protocol*
[14] RTP: *Real Time Protocol*

```
R2 (config-cmap)# exit
R2 (config)# class-map match-all Oro
R2 (config-cmap)# match ip dscp 10 14
R2 (config-cmap)# exit
R2 (config)# class-map match-all Plata
R2 (config-cmap)# match ip dscp 18 22
R2 (config-cmap)# exit
R2 (config)# class-map match-all Bronce
R2 (config-cmap)# match ip dscp 26 30
R2 (config-cmap)# exit
R2 (config)# class-map match-all best-effort
R2 (config-cmap)# match ip dscp 0
R2 (config-cmap)# exit
```

4.3.3. CREACIÓN DE MAPAS DE POLÍTICA

Con los mapas de política, se llevará a cabo el marcado del tráfico, y se establecerá el mecanismo de administración de la congestión. Debe habilitarse un modo de conmutación rápida de los paquetes, denominada CEF, o Cisco Express Forwarding; con esto se puede establecer los mapas de política de entrada a una interfaz en específico. Además, estos mapas de política deben asignarse a las interfaces de los enrutadores. La configuración para el enrutador 1 es la siguiente:

```
R1 (config)# ip cef

R1 (config)# policy-map Politica1
R1 (config-pmap)# class Premium
R1 (config-pmap-c)# priority 800
R1 (config-pmap-c)# exit
R1 (config-pmap)# class Oro
R1 (config-pmap-c)# bandwidth percent 25
R1 (config-pmap-c)# exit
R1 (config-pmap)# class Plata
R1 (config-pmap-c)# bandwidth percent 20
R1 (config-pmap-c)# exit
R1 (config-pmap)# class Bronce
R1 (config-pmap-c)# bandwidth percent 10
R1 (config-pmap-c)# exit
R1 (config-pmap)# class best-effort
R1 (config-pmap-c)# police 56000 1750 1750 conform-action
set-dscp-transmit 0 exceed-action drop violate-action drop
R1 (config-pmap-c)# exit
```

```
R1 (config)# policy-map SetDSCP
R1 (config-pmap)# class Premium
R1 (config-pmap-c)# set ip dscp 46
R1 (config-pmap-c)# exit
R1 (config-pmap)# class Oro
R1 (config-pmap-c)# set ip dscp 10
R1 (config-pmap-c)# set ip dscp 14
R1 (config-pmap-c)# exit
R1 (config-pmap)# class Plata
R1 (config-pmap-c)# set ip dscp 18
R1 (config-pmap-c)# exit
R1 (config-pmap)# class Bronce
R1 (config-pmap-c)# set ip dscp 26
R1 (config-pmap-c)# exit
R1 (config-pmap)# class AF33
R1 (config-pmap-c)# set ip dscp 30
R1 (config-pmap-c)# exit
R1 (config-pmap)# exit
```

Ahora se asignan las políticas creadas a las interfaces del enrutador:

```
R1 (config)# interface gigabitEthernet 0/0
R1 (config-if)# service-policy output Politica1
R1 (config-if)# exit
R1 (config)# interface ethernet 1/0
R1 (config-if)# service-policy input SetDSCP
R1 (config-if)# exit
```

Las configuraciones realizadas para el enrutador 2 son las siguientes:

```
R2 (config)# policy-map Politica-output
R2 (config-pmap)# class Premium
R2 (config-pmap-c)# priority 800
R2 (config-pmap-c)# exit
R2 (config-pmap)# class Oro
R2 (config-pmap-c)# bandwidth percent 25
R2 (config-pmap-c)# random-detect dscp-based
R2 (config-pmap-c)# random-detect dscp 10 20 40 10
R2 (config-pmap-c)# random-detect dscp 14 20 40 20
R2 (config-pmap-c)# exit
R2 (config-pmap)# class Plata
R2 (config-pmap-c)# bandwidth percent 20
R2 (config-pmap-c)# random-detect dscp-based
```

```
R2 (config-pmap-c)# random-detect dscp 18 20 40 10
R2 (config-pmap-c)# random-detect dscp 20 20 40 10
R2 (config-pmap-c)# exit
R2 (config-pmap)# class Bronce
R2 (config-pmap-c)# bandwidth percent 10
R2 (config-pmap-c)# random-detect dscp-based
R2 (config-pmap-c)# random-detect dscp 26 20 40 10
R2 (config-pmap-c)# random-detect dscp 30 20 40 20
R2 (config-pmap-c)# exit
R2 (config-pmap)# exit

R2 (config)# interface gigabitEthernet 0/0
R2 (config-if)# service-policy output Politica-output
R2 (config-if)# exit
R2 (config)# interface ethernet 1/0
R2 (config-if)# service-policy input Politica-output
R2 (config-if)# exit
```

4.4. CONFIGURACIÓN DE ESQUEMA DE QoS PARA IPv6

Continuando con el esquema de configuraciones, las direcciones asignadas a las interfaces de los enrutadores 1 y 2, se encuentran en la Tabla 4.3

Tabla 4.3. Direcciones IPv6 de las interfaces

	Interfaz	Dirección IPv6	Prefijo
Router 1	Gigabit Ethernet 0/0	2001:A:A:A::1	64
	Ethernet 1/0	2001:A:A:B::1	64
Router 2	Gigabit Ethernet 0/0	2001:A:A:A::2	64
	Ethernet 1/0	2001:0:0:1::1	64

Se inicia introduciendo la configuración general en cada enrutador. Para habilitar el enrutamiento IPv6, se utiliza el comando `ipv6 unicast-routing`; con este se habilita automáticamente el protocolo IPv6 en las interfaces de los enrutadores.

```
R1> enable
R1# configure terminal
R1 (config)# ipv6 unicast-routing
R1 (config)# interface ethernet 1/0
R1 (config-if)# ipv6 address 2001:a:a:b::1/64
R1 (config-if)# bandwidth 256
R1 (config-if)# no shutdown
```

```
R1 (config-if)# exit
R1 (config)# interface gigabitEthernet 0/0
R1 (config-if)# ipv6 address 2001:a:a:a::1/64
R1 (config-if)# bandwidth 2048
R1 (config-if)# no shutdown
R1 (config-if)# exit

R2> enable
R2# configure terminal
R2 (config)# ipv6 unicast-routing
R2 (config)# interface ethernet 1/0
R2 (config-if)# ipv6 address 2001:0:0:1::1/64
R2 (config-if)# bandwidth 256
R2 (config-if)# no shutdown
R2 (config-if)# exit
R2 (config)# interface gigabitEthernet 0/0
R2 (config-if)# ipv6 address 2001:a:a:a::2/64
R2 (config-if)# bandwidth 2048
R2 (config-if)# no shutdown
R2 (config-if)# exit
```

Ahora se configure el protocol de enrutamiento, RIPng, de la siguiente manera:

```
R1 (config)# interface gigabitEthernet 0/0
R1 (config-if)# ipv6 rip Routing enable
R1 (config-if)# exit
R1 (config)# interface ethernet 1/0
R1 (config-if)# ipv6 rip Routing enable
R1 (config-if)# exit

R2 (config)# interface gigabitEthernet 0/0
R2 (config-if)# ipv6 rip Routing enable
R2 (config-if)# exit
R2 (config)# interface ethernet 1/0
R2 (config-if)# ipv6 rip Routing enable
R2 (config-if)# exit
```

Se crearán las mismas listas de acceso establecidas en 4.3.1, y sus correspondientes grupos asignados a un valor DSCP, siguiendo los mismos comandos de configuración. Esta puede verificarse en 4.3.2.

4.4.1. CONFIGURACIÓN DE MAPAS DE CLASE

Los mapas de clase que se crearán serán los mismos establecidos en el esquema de calidad de servicio en 3.4. Los comandos de configuración son básicamente los mismos. Tal configuración es la siguiente:

```
R1 (config)# class-map match-all Premium
R1 (config-cmap)# match dscp 46
R1 (config-cmap)# match access-group 101
R1 (config-cmap)# exit
R1 (config)# class-map match-all Oro
R1 (config-cmap)# match dscp 10 14
R1 (config-cmap)# match protocol rtsp
R1 (config-cmap)# match protocol rtp
R1 (config-cmap)# exit
R1 (config)# class-map match-all Plata
R1 (config-cmap)# match dscp 18 22
R1 (config-cmap)# match access-group 109
R1 (config-cmap)# match protocol secure-http
R1 (config-cmap)# match protocol secure-ftp
R1 (config-cmap)# match protocol ipsec
R1 (config-cmap)# exit
R1 (config)# class-map match-all Bronce
R1 (config-cmap)# match dscp 26 30
R1 (config-cmap)# match protocol dhcp
R1 (config-cmap)# match protocol icmp
R1 (config-cmap)# match protocol snmp
R1 (config-cmap)# match access-group 108
R1 (config-cmap)# match access-group 110
R1 (config-cmap)# exit
R1 (config)# class-map match-all best-effort
R1 (config-cmap)# match access-group 105
R1 (config-cmap)# exit
```

Para el caso del enrutador 2, se realiza la siguiente configuración:

```
R2 (config)# class-map match-all Premium
R2 (config-cmap)# match dscp 46
R2 (config-cmap)# exit
R2 (config)# class-map match-all Oro
R2 (config-cmap)# match dscp 10 14
R2 (config-cmap)# exit
R2 (config)# class-map match-all Plata
```

```
R2 (config-cmap)# match dscp 18 22
R2 (config-cmap)# exit
R2 (config)# class-map match-all Bronce
R2 (config-cmap)# match dscp 26 30
R2 (config-cmap)# exit
R2 (config)# class-map match-all best-effort
R2 (config-cmap)# match dscp 0
R2 (config-cmap)# exit
```

4.4.2. CONFIGURACIÓN DE MAPAS DE POLÍTICA

En 4.3.3, se crearon y asignaron los mapas de política a las interfaces de los enrutadores. Tales políticas creadas pueden volver a ser utilizadas en la configuración de IPv6, bajo los mismos comandos.

4.5. CONFIGURACIÓN DE IP SLA

IP SLA es una función de los IOS de Cisco, que permite analizar un acuerdo de nivel de servicios (SLA) para una aplicación o servicio IP. Se crearan varias instancias para comprobar los parámetros de QoS, cada una se dará para diferentes tipos de tráfico. La configuración se mantiene tanto para IPv4 como para IPv6, con la diferencia de las direcciones de las interfaces. Se configuran las pruebas para los siguientes niveles de servicio:

```
! UDP jitter
R1 (config)# ip sla 1
R1 (config-ip-sla)# udp-jitter 10.10.12.2/2001:a:a:a::2 16838
source-ip 192.168.12.1/2001:a:a:b::1
R1 (config-ip-sla-jitter)# threshold 500
R1 (config-ip-sla-jitter)# timeout 500
R1 (config-ip-sla-jitter)# frequency 15
R1 (config-ip-sla-jitter)# tos 56 / traffic-class 56
R1 (config-ip-sla-jitter)# exit
R1 (config)# ip sla schedule 1 start-time now life forever

! VoIP UDP jitter G.711ulaw
R1 (config)# ip sla 2
R1 (config-ip-sla)# udp-jitter 10.10.12.2/2001:a:a:a::2 16834
source-ip 192.168.12.1/2001:a:a:b::1 codec g711ulaw
R1 (config-ip-sla-jitter)# threshold 500
R1 (config-ip-sla-jitter)# timeout 500
R1 (config-ip-sla-jitter)# frequency 25
R1 (config-ip-sla-jitter)# tos 184 / traffic-class 184
R1 (config-ip-sla-jitter)# exit
R1 (config)# ip sla schedule 2 start-time now life forever
```

```
! VoIP UDP jitter G.729a
R1 (config)# ip sla  3
R1 (config-ip-sla)# udp-jitter 10.10.12.2/2001:a:a:a::2 16836
source-ip 192.168.12.1/2001:a:a:b::1 codec g729a
R1 (config-ip-sla-jitter)# threshold 500
R1 (config-ip-sla-jitter)# timeout 500
R1 (config-ip-sla-jitter)# frequency 25
R1 (config-ip-sla-jitter)# tos 184 / traffic-class 184
R1 (config-ip-sla-jitter)# exit
R1 (config)# ip sla schedule 3 start-time now life forever

! UDP Echo
R1 (config)# ip sla 4
R1 (config-ip-sla)# udp-echo 10.10.12.2/2001:a:a:a::2 16835 source-
ip 192.168.12.1/2001:a:a:b::1
R1 (config-ip-sla-udp)# threshold 500
R1 (config-ip-sla-udp)# timeout 500
R1 (config-ip-sla-udp)# frequency 10
R1 (config-ip-sla-udp)# tos 184 / traffic-class 184
R1 (config-ip-sla-udp)# exit
R1 (config)# ip sla schedule 4 start-time now life forever


! TCPconnect Telnet
R1 (config)# ip sla 5
R1 (config-ip-sla)# tcp-connect 10.10.12.2/2001:a:a:a::2 23 source-
ip 192.168.12.1/2001:a:a:b::1
R1 (config-ip-sla-tcp)# threshold 500
R1 (config-ip-sla-tcp)# timeout 500
R1 (config-ip-sla-tcp)# frequency 20
R1 (config-ip-sla-tcp)# tos 104 / traffic-class 104
R1 (config-ip-sla-tcp)# exit
R1 (config)# ip sla schedule 5 start-time now life forever

! TCPconnect FTP
R1 (config)# ip sla 6
R1 (config-ip-sla)# tcp-connect 10.10.12.2/2001:a:a:a::2 21 source-
ip 192.168.12.1/2001:a:a:b::1
R1 (config-ip-sla-tcp)# threshold 500
R1 (config-ip-sla-tcp)# timeout 500
R1 (config-ip-sla-tcp)# frequency 20
R1 (config-ip-sla-tcp)# exit
R1 (config)# ip sla schedule 6 start-time now life forever

! TCPconnect HTTPS
R1 (config)# ip sla 7
```

```
R1 (config-ip-sla)# tcp-connect 10.10.12.2/2001:a:a:a::2 443 source-
ip 192.168.12.1/2001:a:a:b::1
R1 (config-ip-sla-tcp)# threshold 500
R1 (config-ip-sla-tcp)# timeout 500
R1 (config-ip-sla-tcp)# frequency 20
R1 (config-ip-sla-tcp)# tos 88 / traffic-class 88
R1 (config-ip-sla-tcp)# exit
R1 (config)# ip sla schedule 7 start-time now life forever

! TCPconnect SSH
R1 (config)# ip sla 8
R1 (config-ip-sla)# tcp-connect 10.10.12.2/2001:a:a:a::2 22 source-
ip 192.168.12.1/2001:a:a:b::1
R1 (config-ip-sla-tcp)# threshold 500
R1 (config-ip-sla-tcp)# timeout 500
R1 (config-ip-sla-tcp)# frequency 20
R1 (config-ip-sla-tcp)# tos 104 / traffic-class 104
R1 (config-ip-sla-tcp)# exit
R1 (config)# ip sla schedule 8 start-time now life forever

! ICMP Echo
R1 (config)# ip sla 9
R1 (config-ip-sla)# icmp-echo 10.10.12.2/2001:a:a:a::2 source-
interface Ethernet 1/0
R1 (config-ip-sla-echo)# threshold 500
R1 (config-ip-sla-echo)# timeout 500
R1 (config-ip-sla-echo)# frequency 10
R1 (config-ip-sla-echo)# tos 120 / traffic-class 120
R1 (config-ip-sla-echo)# exit
R1 (config)# ip sla schedule 9 start-time now life forever
```

4.6. VERIFICACIÓN DE CONFIGURACIONES EN SIMULADOR

Como se ha mencionado previamente, GNS3 es un emulador se hardware, y ofrece una interfaz de configuración por línea de comandos, para las versiones de IOS instaladas en los equipos, a través de la cual se ingresaran las configuraciones correspondientes al esquema diseñado, que fueron mostradas en las secciones 4.3, 4.4 y 4.5. A continuación se mostrará la verificación de las configuraciones realizadas.

4.6.1. IPV4

A continuación se muestran imágenes correspondientes a la verificación de las configuraciones de la red IPv4.

```
R1#sh ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0/0                unassigned      YES NVRAM  administratively down down
GigabitEthernet0/0         10.10.12.1      YES NVRAM  up                    up
Ethernet1/0                192.168.12.1    YES NVRAM  up                    up
Ethernet1/1                unassigned      YES NVRAM  administratively down down
Ethernet1/2                unassigned      YES NVRAM  administratively down down
Ethernet1/3                unassigned      YES NVRAM  administratively down down
R1#
```

Figura 4.3. Configuración de interfaces

```
R1#sh interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is i82543 (Livengood), address is ca00.1c38.0008 (bia ca00.1c38.0008)
  Internet address is 10.10.12.1/24
  MTU 1500 bytes, BW 2048 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s, link type is autonegotiation, media type is SX
  output flow-control is XON, input flow-control is XON
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: Class-based queueing
  Output queue: 0/1000/0 (size/max total/drops)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2285 packets input, 213182 bytes, 0 no buffer
     Received 809 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     2289 packets output, 213445 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     4 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

Figura 4.4. Interfaz Gigabit Ethernet 0/0

```
R1#sh interface Ethernet 1/0
Ethernet1/0 is up, line protocol is up
  Hardware is AmdP2, address is ca00.1c38.001c (bia ca00.1c38.001c)
  Internet address is 192.168.12.1/24
  MTU 1500 bytes, BW 256 Kbit/sec, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 input packets with dribble condition detected
     2297 packets output, 223190 bytes, 0 underruns
     0 output errors, 0 collisions, 18 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Figura 4.5. Interfaz Ethernet 1/0

```
R1#sh ip protocol
*** IP Routing is NSF aware ***

Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 23 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface             Send  Recv  Triggered RIP  Key-chain
    GigabitEthernet0/0    1     1 2
    Ethernet1/0           1     1 2
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    192.168.12.0
    200.172.16.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.10.12.2           120      00:00:28
  Distance: (default is 120)

R1#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.10.12.0/24 is directly connected, GigabitEthernet0/0
L        10.10.12.1/32 is directly connected, GigabitEthernet0/0
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, Ethernet1/0
L        192.168.12.1/32 is directly connected, Ethernet1/0
R     200.172.16.0/24 [120/1] via 10.10.12.2, 00:00:24, GigabitEthernet0/0
```

Figura 4.6. Protocolo de enrutamiento y Rutas IP

```
R1#sh access-lists
Extended IP access list 101
    10 permit udp any any range 16384 32768
Extended IP access list 102
    10 permit tcp any any range ftp telnet
Extended IP access list 103
    10 permit tcp any any eq tacacs
Extended IP access list 104
    10 permit tcp any any eq www
Extended IP access list 105
    10 permit ip any any (459 matches)
Extended IP access list 108
    10 permit tcp any any eq telnet
Extended IP access list 109
    10 permit tcp any any eq 443
Extended IP access list 110
    10 permit icmp any any
```

Figura 4.7. Listas de acceso

```
R1#sh class-map
 Class Map match-all best-effort (id 5)
   Match access-group  105

 Class Map match-any class-default (id 0)
   Match any

 Class Map match-all Bronce (id 4)
   Match ip  dscp af31 (26) af33 (30)
   Match protocol dhcp
   Match protocol icmp
   Match protocol snmp
   Match access-group  108
   Match access-group  110

 Class Map match-all Plata (id 3)
   Match ip  dscp af21 (18) af23 (22)
   Match protocol secure-http
   Match protocol secure-ftp
   Match protocol ipsec
   Match access-group  109

 Class Map match-all Oro (id 2)
   Match ip  dscp af11 (10) af13 (14)
   Match protocol rtsp
   Match protocol rtp

 Class Map match-all Premium (id 1)
   Match ip  dscp ef (46)
   Match access-group  101
```

Figura 4.8. Mapas de clase configurados

```
R1#sh policy-map
  Policy Map Politica1
    Class Premium
      priority 800 (kbps)
    Class Oro
      bandwidth 25 (%)
    Class Plata
      bandwidth 20 (%)
    Class Bronce
      bandwidth 10 (%)
    Class best-effort
     police cir 56000 bc 1750 be 1750
       conform-action set-dscp-transmit default
       exceed-action drop
       violate-action drop

  Policy Map SetDSCP
    Class Premium
      set dscp ef
    Class Oro
      set dscp af13
    Class Plata
      set dscp af21
    Class Bronce
      set dscp af31
```

Figura 4.9. Mapas de política

```
R1#sh policy-map interface gigabitEthernet 0/0
 GigabitEthernet0/0

  Service-policy output: Politica1

    queue stats for all priority classes:

      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0

    Class-map: Premium (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp ef (46)
      Match: access-group 101
      Priority: 800 kbps, burst bytes 20000, b/w exceed drops: 0


    Class-map: Oro (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af11 (10) af13 (14)
      Match: protocol rtsp
      Match: protocol rtp
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth 25% (512 kbps)

    Class-map: Plata (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af21 (18) af23 (22)
      Match: protocol secure-http
      Match: protocol secure-ftp
      Match: protocol ipsec
      Match: access-group 109
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth 20% (409 kbps)

    Class-map: Bronce (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af31 (26) af33 (30)
      Match: protocol dhcp
      Match: protocol icmp
      Match: protocol snmp
      Match: access-group 108
      Match: access-group 110
      Queueing
      queue limit 64 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 0/0
      bandwidth 10% (204 kbps)
Class-map: best-effort (match-all)
  77 packets, 9759 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: access-group 105
  police:
      cir 56000 bps, bc 1750 bytes, be 1750 bytes
    conformed 77 packets, 5082 bytes; actions:
      set-dscp-transmit default
    exceeded 0 packets, 0 bytes; actions:
      drop
    violated 0 packets, 0 bytes; actions:
      drop
    conformed 0 bps, exceed 0 bps, violate 0 bps

Class-map: class-default (match-any)
  255 packets, 25844 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any

  queue limit 64 packets
  (queue depth/total drops/no-buffer drops) 0/0/0
  (pkts output/bytes output) 332/30926
```

Figura 4.10. Política asociada a interfaz Gigabit Ethernet 0/0

```
R1#sh policy-map interface Ethernet 1/0
 Ethernet1/0

  Service-policy input: SetDSCP

    Class-map: Premium (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp ef (46)
      Match: access-group 101
      QoS Set
        dscp ef
          Packets marked 0

    Class-map: Oro (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af11 (10) af13 (14)
      Match: protocol rtsp
      Match: protocol rtp
      QoS Set
        dscp af13
          Packets marked 0

    Class-map: Plata (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af21 (18) af23 (22)
      Match: protocol secure-http
      Match: protocol secure-ftp
      Match: protocol ipsec
      Match: access-group 109
      QoS Set
        dscp af21
          Packets marked 0

    Class-map: Bronce (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af31 (26) af33 (30)
      Match: protocol dhcp
      Match: protocol icmp
      Match: protocol snmp
      Match: access-group 108
      Match: access-group 110
      QoS Set
        dscp af31
          Packets marked 0

    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Figura 4.11. Política asociada a interfaz Ethernet 1/0

A continuación se muestran las configuraciones realizadas de cada uno de los IP SLA. De la Figura 4.12 a la Figura 4.20, se muestran las configuraciones de los IP SLA1 al 9, respectivamente.

```
R1#sh ip sla configuration 1
IP SLAs, Infrastructure Engine-II.
Entry number: 1
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 10.10.12.2/192.168.12.1
Target port/Source port: 16838/0
Type Of Service parameter: 0x38
Request size (ARR data portion): 32
Operation timeout (milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
Type Of Service parameters: 0x38
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 15  (not considered if randomly scheduled)
   Next Scheduled Start Time: Pending trigger
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
```

Figura 4.12. Configuración de IP SLA 1

```
R1#sh ip sla configuration 2
IP SLAs, Infrastructure Engine-II.
Entry number: 2
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 10.10.12.2/192.168.12.1
Target port/Source port: 16834/0
Type Of Service parameter: 0xB8
Operation timeout (milliseconds): 500
Codec Type: g711ulaw
Codec Number Of Packets: 1000
Codec Packet Size: 172
Codec Interval (milliseconds): 20
Advantage Factor: 0
Type Of Service parameters: 0xB8
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 25  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
```

Figura 4.13. Configuración de IP SLA 2

```
R1#sh ip sla configuration 3
IP SLAs, Infrastructure Engine-II.
Entry number: 3
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 10.10.12.2/192.168.12.1
Target port/Source port: 16836/0
Type Of Service parameter: 0xB8
Operation timeout (milliseconds): 500
Codec Type: g729a
Codec Number Of Packets: 1000
Codec Packet Size: 32
Codec Interval (milliseconds): 20
Advantage Factor: 0
Type Of Service parameters: 0xB8
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 25  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): Forever
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
```

Figura 4.14. Configuración IP SLA 3

```
R1#sh ip sla configuration 4
IP SLAs, Infrastructure Engine-II.
Entry number: 4
Owner:
Tag:
Type of operation to perform: udp-echo
Target address/Source address: 10.10.12.2/192.168.12.1
Target port/Source port: 16835/0
Type Of Service parameter: 0x28
Request size (ARR data portion): 16
Operation timeout (milliseconds): 500
Type Of Service parameters: 0x28
Verify data: No
Data pattern:
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly scheduled)
   Next Scheduled Start Time: Pending trigger
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
```

Figura 4.15. Configuración IP SLA 4

```
R1#sh ip sla configuration 5
IP SLAs, Infrastructure Engine-II.
Entry number: 5
Owner:
Tag:
Type of operation to perform: tcp-connect
Target address/Source address: 10.10.12.2/192.168.12.1
Target port/Source port: 23/0
Type Of Service parameter: 0x68
Operation timeout (milliseconds): 60000
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Pending trigger
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:
```

Figura 4.16. Configuración IP SLA 5

```
R1#sh ip sla configuration 6
IP SLAs, Infrastructure Engine-II.
Entry number: 6
Owner:
Tag:
Type of operation to perform: tcp-connect
Target address/Source address: 10.10.12.2/192.168.12.1
Target port/Source port: 21/0
Type Of Service parameter: 0x0
Operation timeout (milliseconds): 60000
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Pending trigger
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:
```

Figura 4.17. Configuración IP SLA 6

```
R1#sh ip sla configuration 7
IP SLAs, Infrastructure Engine-II.
Entry number: 7
Owner:
Tag:
Type of operation to perform: tcp-connect
Target address/Source address: 10.10.12.2/192.168.12.1
Target port/Source port: 443/0
Type Of Service parameter: 0x58
Operation timeout (milliseconds): 60000
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Pending trigger
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:
```

Figura 4.18. Configuración IP SLA 7

```
R1#sh ip sla configuration 8
IP SLAs, Infrastructure Engine-II.
Entry number: 8
Owner:
Tag:
Type of operation to perform: tcp-connect
Target address/Source address: 10.10.12.2/192.168.12.1
Target port/Source port: 22/0
Type Of Service parameter: 0x68
Operation timeout (milliseconds): 60000
Vrf Name:
Control Packets: enabled
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Pending trigger
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:
```

Figura 4.19. Configuración IP SLA 8

```
R1#sh ip sla configuration 9
IP SLAs, Infrastructure Engine-II.
Entry number: 9
Owner:
Tag:
Type of operation to perform: icmp-echo
Target address/Source interface: 10.10.12.2/Ethernet1/0
Type Of Service parameter: 0x78
Request size (ARR data portion): 28
Operation timeout (milliseconds): 500
Verify data: No
Vrf Name:
Schedule:
   Operation frequency (seconds): 10  (not considered if randomly scheduled)
   Next Scheduled Start Time: Pending trigger
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 500 (not considered if react RTT is configured)
Distribution Statistics:
   Number of statistic hours kept: 2
   Number of statistic distribution buckets kept: 1
   Statistic distribution interval (milliseconds): 20
History Statistics:
   Number of history Lives kept: 0
   Number of history Buckets kept: 15
   History Filter Type: None
Enhanced History:
```

Figura 4.20. Configuración IP SLA 9

4.6.2. IPv6

En esta sección se muestran las configuraciones realizadas para el caso de la red IPv6.

```
R1#sh interface gigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  Hardware is i82543 (Livengood), address is ca0b.1eac.0008 (bia ca0b.1eac.0008)
  MTU 1500 bytes, BW 2048 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s, link type is autonegotiation, media type is SX
  output flow-control is XON, input flow-control is XON
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:02:17, output 00:02:13, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: Class-based queueing
  Output queue: 0/1000/0 (size/max total/drops)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     2370 packets input, 801477 bytes, 0 no buffer
     Received 2370 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     0 input packets with dribble condition detected
     10325 packets output, 938745 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     7 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
R1#
```

Figura 4.21. Configuración interfaz Gigabit Ethernet 0/0

```
#sh interface ethernet 0/0
hernet0/0 is administratively down, line protocol is down
Hardware is i82543 (Livengood), address is ca0b.1eac.0006 (bia ca0b.1eac.0006)
MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
   reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 no buffer
   Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
   0 input packets with dribble condition detected
   0 packets output, 0 bytes, 0 underruns
   0 output errors, 0 collisions, 0 interface resets
   0 unknown protocol drops
   0 babbles, 0 late collision, 0 deferred
   0 lost carrier, 0 no carrier
   0 output buffer failures, 0 output buffers swapped out
```

Figura 4.22. Configuración interfaz Ethernet 1/0

```
R1#sh ipv6 protocol
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "rip Routing"
  Interfaces:
    Ethernet1/0
    GigabitEthernet0/0
  Redistribution:
    None
R1#sh ipv6 route
IPv6 Routing Table - default - 5 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external, ND - Neighbor Discovery
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C   2001:A:A:A::/64 [0/0]
     via GigabitEthernet0/0, directly connected
L   2001:A:A:A::1/128 [0/0]
     via GigabitEthernet0/0, receive
C   2001:A:A:B::/64 [0/0]
     via Ethernet1/0, directly connected
L   2001:A:A:B::1/128 [0/0]
     via Ethernet1/0, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

Figura 4.23. Protocolo de enrutamiento

```
R1#sh class-map
 Class Map match-all best-effort (id 5)
   Match access-group  105

 Class Map match-any class-default (id 0)
   Match any

 Class Map match-all Bronce (id 4)
   Match   dscp af31 (26) af33 (30)
   Match protocol dhcp
   Match protocol icmp
   Match protocol snmp
   Match access-group  108
   Match access-group  110

 Class Map match-all Plata (id 3)
   Match   dscp af21 (18) af23 (22)
   Match access-group  109
   Match protocol secure-http
   Match protocol secure-ftp
   Match protocol ipsec

 Class Map match-all Oro (id 2)
   Match   dscp af11 (10) af13 (14)
   Match protocol rtsp
   Match protocol rtp

 Class Map match-all Premium (id 1)
   Match   dscp ef (46)
   Match access-group  101
```

Figura 4.24. Mapas de clase

```
R1#sh policy-map
 Policy Map Politica1
   Class Premium
     priority 800 (kbps)
   Class Oro
     bandwidth 25 (%)
   Class Plata
     bandwidth 20 (%)
   Class Bronce
     bandwidth 10 (%)
   Class best-effort
    police cir 56000 bc 1750 be 1750
      conform-action set-dscp-transmit default
      exceed-action drop
      violate-action drop

 Policy Map SetDSCP
   Class Premium
     set dscp ef
   Class Oro
     set dscp af13
   Class Plata
     set dscp af21
   Class Bronce
     set dscp af31
```

Figura 4.25. Mapas de política

4.7. ESTADÍSTICAS DE TRÁFICO GENERADO

Siguiendo la metodología de generación de tráfico explicada en 4.5, se dejaron corriendo las pruebas por tiempo aproximado de 1 hora, y se recolectaron las estadísticas correspondientes a los IPSLA configurados; obteniendo datos como latencia en una dirección, de origen a destino, y de destino a origen; latencia de ida y vuelta, paquetes perdidos, y jitter.

Las estadísticas se recopilaron a través del comando `show ip sla statistics`, en el modo de configuración privilegiado.

Las estadísticas (más relevantes) recopiladas para los IP SLA 1, 2 y 3, se encuentran registrados en la Tabla 4.4 y la Tabla 4.5. En ellas se pueden apreciar los valores tanto para IPv4 como para IPv6. Del mismo modo, la Tabla 4.6 recopila las estadísticas tomadas para los IP SLA 4, 5, 6, 7, 8 y 9. Se tabularon en tablas diferentes, debido a que la configuración realizada para los 3 primeros IP SLA, arrojan datos distintos a las configuraciones del resto de las pruebas.

Tabla 4.4. Estadísticas IP SLA 1, 2 y 3 (a)

Estadísticas	IP SLA 1		IP SLA 2		IP SLA 3	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
Pérdida de paquetes SD	0	0	0	0	0	0
Pérdida de paquetes DS	1	0	0	0	0	0
Llegada tardía	0	0	0	0	0	0
TAIL DROP	0	0	349	0	313	0
Fuera de secuencia	0	0	0	0	0	0
ICPIF	0	0	1	1	10	10
MOS	0	0	4,34	4,34	4,06	4,06

Tabla 4.5. Estadísticas IP SLA 1, 2 y 3 (b)

Estadísticas	IP SLA 1		IP SLA 2		IP SLA 3	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
Número de RTT	9	10	651	1000	687	995
RTT promedio (ms)	72	50	140	93	109	68
Muestras Jitter SD	8	9	632	999	669	994
Muestras Jitter DS	8	9	624	999	669	994
Jitter promedio SD (ms)	35	20	44	24	40	20
Jitter promedio DS (ms)	11	12	37	21	31	17
Muestras latencia OW[15]	6	2	11	401	14	574
Latencia OW promedio SD (ms)	43	4	76	63	116	39
Latencia OW promedio DS (ms)	20	58	85	69	19	48

[15] OW corresponde a una dirección (One-Way). Latencia en una dirección.

Tabla 4.6. Estadísticas IP SLA 4, 5, 6, 7, 8 y 9

Estadísticas	IP SLA 4		IP SLA 5		IP SLA 6		IP SLA 7		IP SLA 8		IP SLA 9	
	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6	IPv4	IPv6
RTT promedio (ms)	70	56	92	80	92	80	96	72	100	72	56	32

4.7.1. ANÁLISIS DE RESULTADOS

Como medida de calidad de servicio en una red, deben considerarse la tasa de pérdida de paquetes, los retardos y las variaciones del retardo.

Los tráficos de IP SLA fueron generados simultáneamente, para que cada tipo de tráfico compitiera por el ancho de banda disponible y se hiciera uso de los métodos de encolamiento y evasión de la congestión. De acuerdo a la frecuencia configurada para los IP SLA del 5 al 9, se percibe que el tráfico generado en estos casos es mayor, para probar si la calidad del tráfico de voz fue degradada. Esto quiere decir que para el mismo tiempo de simulación, se generaron más ramas de tráfico tcp que udp.

Es necesario resaltar, que el comportamiento que se presentó en la red, se vio influido por el rendimiento del equipo en el cual se estaba corriendo el simulador, debido a que este utiliza los recursos físicos del mismo para poder emular las IOS de los enrutadores.

La Figura 4.26 muestra el comportamiento que presentó la medida de RTT para todos los IP SLA. Se puede apreciar un comportamiento muy similar entre los diferentes tipos de tráfico generados, tanto para IPv4 como para IPv6, pero obteniendo valores menores para todos los casos, en las medidas de IPv6. El valor máximo obtenido fue de 140 ms, correspondiente al IP SLA 2 de IPvr4 (que genera tráfico similar al de una conversación g711ulaw); y el valor mínimo fue de 32 ms, correspondiente al IP SLA 9 de IPv6 (toma las mediciones del tráfico ICMP-echo).

La Figura 4.27 y la Figura 4.28, muestran el comportamiento presente para la medida de latencia en una dirección, respectivamente de origen a destino (SD), y de destino a origen (DS), para los IP SLA 1, 2 y 3.

En el caso de la latencia OW SD[16], se presento un mejor comportamiento dentro de la red IPv6, con un valor mínimo de 4 ms en el IP SLA 1 (medida de UDP-jitter), y un valor máximo de 63 ms en el IP SLA 2 (medida de UDP-Jitter g711ulaw). Para IPv4 se obtuvo un valor máximo de 116 ms, para el IP SLA 3 (medida de UDP-Jitter g729a).

[16] Latencia OW SD, equivale a decir Latencia One-way Source-destination (una dirección origen-destino)

En la latencia OW DS[17], la red IPv6 mantuvo valores muy estables. Y la red IPv4 tuvo menores medidas de retardo de destino a origen.

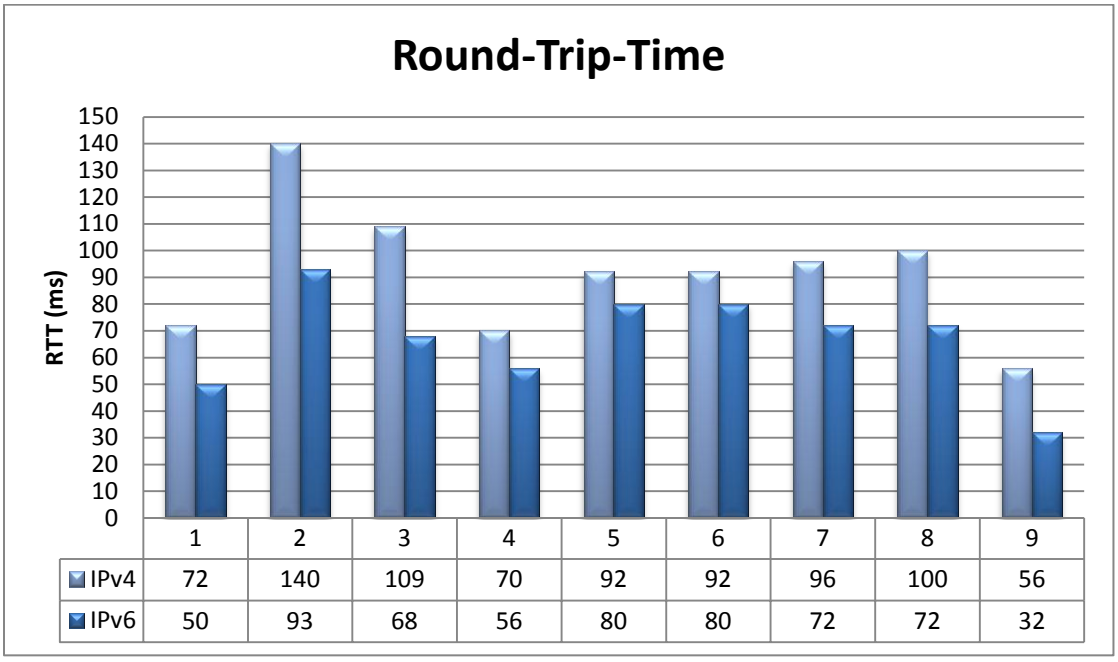

Figura 4.26.Round-Trip-Time promedio para cada IP SLA

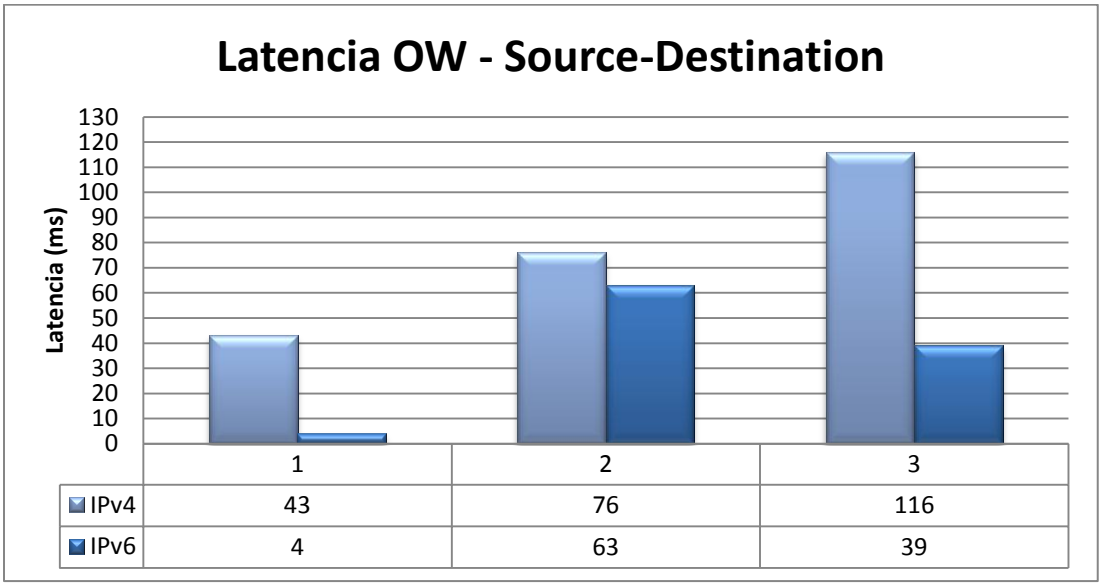

Figura 4.27. Medida de latencia en una dirección Origen-Destino para los IP SLA 1, 2 y 3

[17] Latencia OW DS, equivale a decir Latencia One-way Destination-source (una dirección destino-origen)

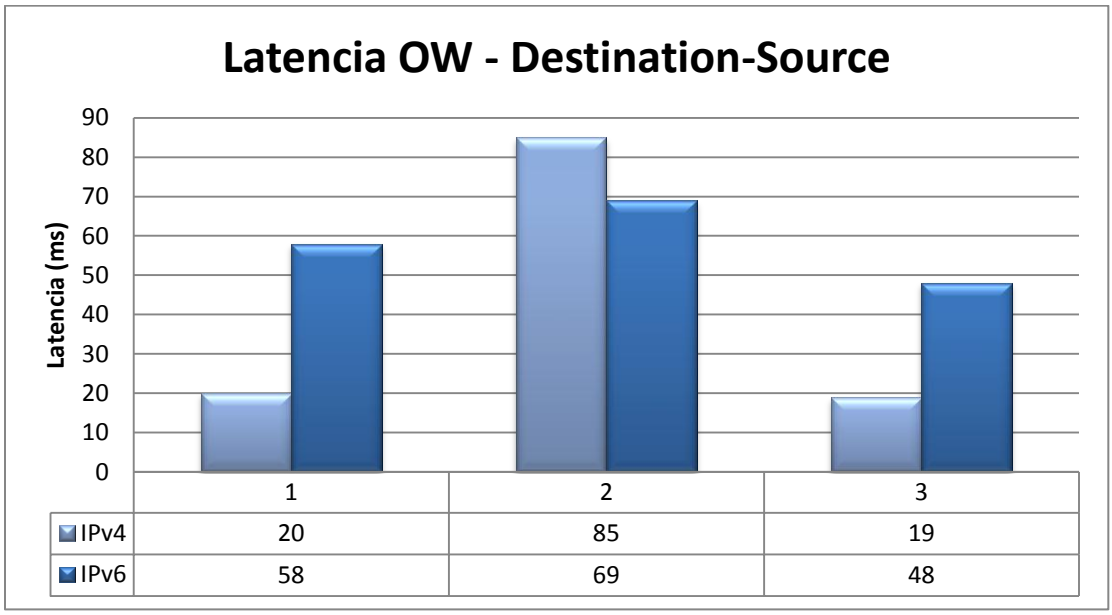

Figura 4.28. . Medida de latencia en una dirección Destino-Origen para los IP SLA 1, 2 y 3

En la Figura 4.29 y Figura 4.30, se aprecian los tiempos de Jitter SD y DS en las redes IPv4 e IPv6, tomados de los IP Sla 1, 2 y 3 (tráfico UDP). También puede apreciarse que los tiempos fueron menores en la red IPv6 que en la red IPv4.

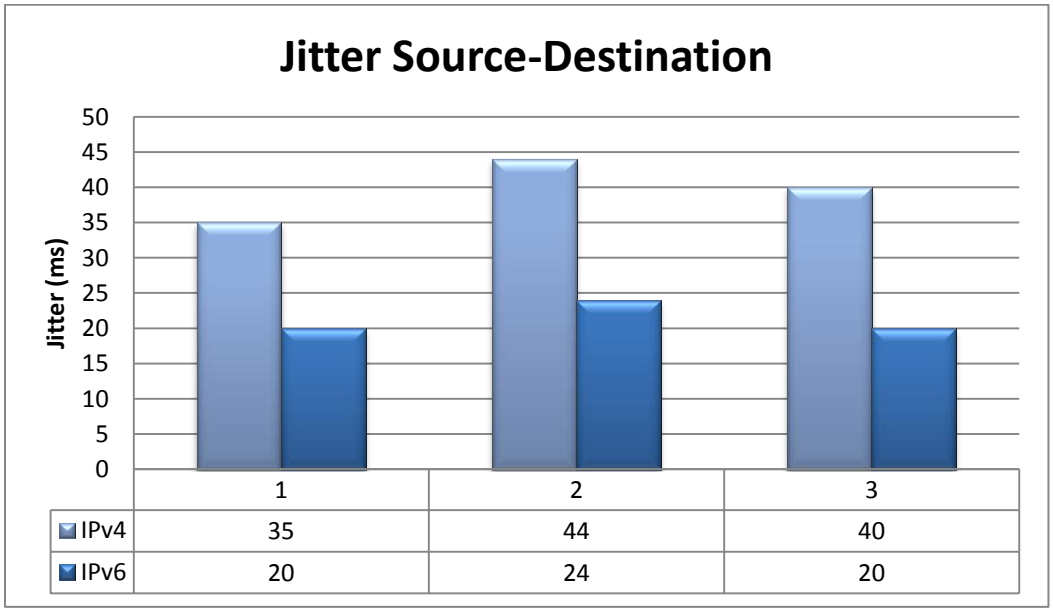

Figura 4.29. Medida de Jitter de origen a destino para los IP SLA 1, 2 y 3

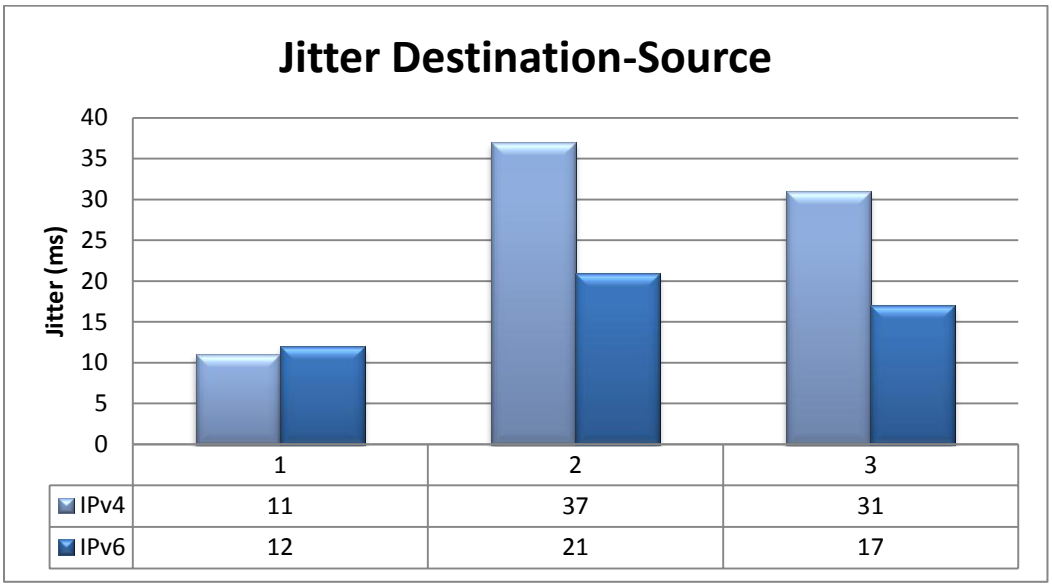

Figura 4.30. Medida de Jitter de destino a origen para los IP SLA 1, 2 y 3

En cuanto a la pérdida de paquetes, de acuerdo a la Tabla 4.4, se puede apreciar que para el tráfico UDP (IP SLA 1, 2 y 3), que fue donde se obtuvieron estadísticas relacionadas, en la red IPv6 no se obtuvieron ningún tipo de pérdidas ni inconvenientes. Pero en la red IPv4, se obtuvieron algunos errores, teniendo ciertos descartes de paquetes en el IP SLA 2 y 3.

En IPv6, aun cuando los paquetes son de mayor tamaño, su procesamiento por los enrutadores se ha vuelto más sencillo que el procesamiento de los paquetes IPv4, debido a que fueron eliminados campos de su cabecera, dejando solo los campos realmente usados. Esto puede conllevar a una mejor respuesta de los tiempos de retardo y de jitter en comparación con IPv4.

De acuerdo a la Tabla 3.1. Parámetros por clase de servicio, se verifica si los datos obtenidos caen dentro de los requisitos de QoS que debe tener cada tipo de tráfico.

El tráfico, de los que fue generado, más sensible a las características del canal, es el de voz. Este cae dentro de la clase Premium, y debe tener un retardo ≤150 ms; jitter ≤30ms; latencia OW ≤ 25ms. Para los IP SLA 1, 2, y 3 en IPv6, se puede ver que estos parámetros caen dentro de lo establecido. Para el tráfico en IPv4, se puede ver que el valor de jitter sobrepasa un poco lo establecido, pero no significativamente. El valor de RTT cae dentro del rango establecido en ambas situaciones, y el porcentaje de paquetes entregados también cae dentro de lo que se debe cumplir.

Adicionalmente, para el trafico de voz, se tienen dos medidas más: ICPIF y MOS.

ICPIF intenta cuantificar, para propósitos de comparación y planeación, el factor de deterioro de la calidad de la voz encontrado dentro de la red. La Tabla 4.7 muestra la calidad conversacional de acuerdo a este valor.

El IP SLA 2, que corresponde a una conversación de códec g711, tiene un ICPIF de 1, que equivale a una calidad de conversación muy buena. Mientras que el IP SLA 3, conversación de códec g729a tiene un ICPIF de 120, cuya calidad de conversación es buena.

Tabla 4.7. Niveles de calidad de acuerdo a ICPIF

Valor máximo de ICPIF	Calidad de conversación
5	Muy bueno
10	Bueno
20	Adecuado
30	Caso limitado
45	Caso limitado excepcional
55	Mala reacción por parte de usuarios

Cada códec utilizado para transmisión de VoIP provee cierto nivel de calidad. MOS (Mean Opinion Score) es comúnmente utilizado para determinar la calidad del sonido producido por un códec en específico. La Tabla 4.8 muestra los valores de MOS, y que percepción tienen los usuarios de las muestras de las conversaciones según el códec usado.

De acuerdo a esto, se puede apreciar que el códec g711 presentó un MOS de 4.34 y el g729a de 4.06, cayendo ambos dentro de una calidad buena, cuya degradación prácticamente no se percibe, y no es molesta durante su uso.

Tabla 4.8. Rango MOS

Valor	Calidad	Descripción de degradación de calidad
5	Excelente	Imperceptible
4	Buena	Meramente percibida, pero no molesta
3	Justa	Percibida y un poco molesta
2	Pobre	Molesta pero no acusada
1	Mala	Muy molesta y acusable

5. CONCLUSIONES

- Para poder establecer un esquema de calidad de servicio, se deben determinar los parámetros comprometidos dentro de los acuerdos de nivel de servicio que se apliquen; puesto que se tiene claro cuales son los niveles que se desean alcanzar, y los tráficos a transmitir.
- Los procedimientos de calidad de servicio para el control y la evasión de la congestión, mejoran considerablemente el rendimiento de una red al controlar parámetros como ancho de banda, latencia, variación de retardos y tasa de paquetes perdidos; evitando la disminución del nivel de servicio ofrecido.
- Al establecer un esquema de QoS dentro de una red, se pretende una justa asignación de recursos para los servicios y aplicaciones que circulan a través de la misma, de acuerdo a los requerimientos que presente cada uno. Inclusive dentro de periodos de congestión.
- Gracias a que los mecanismos (encolamiento y evasión de congestión) y las arquitecturas de servicio desarrolladas hasta el momento son bastante robustas, es posible implementarlas para proveer calidad de servicio en una red IPv4 como en una red IPv6.
- El modelo de arquitectura de DiffServ, provee servicios diferenciados, con los cuales, luego del marcado de los paquetes (en este caso, a través de DSCP), los mecanismos de manipulación y evasión de la congestión pueden actuar sobre el tráfico circulante en la red.
- A pesar de que el formato de un paquete IPv6 es de mayor tamaño que el paquete en IPv4, el procesamiento de los mismos por los enrutadores, es más eficiente en un paquete

IPv6. Debido a que fueron eliminados muchos campos en la cabecera que no eran utilizados.

- Ipv6 posee campos adicionales que colaboran con la aplicación de calidad de servicio sobre su red, como lo es el campo de etiqueta de flujo, con el que se hace la distinción de los flujos de tráfico circulantes en la red, facilitando su interpretación, y manipulación.
- Se obtuvo un mejor rendimiento en los tiempos de retardo, jitter, y tasa de pérdida de paquetes dentro de la red IPv6.
- El esquema de calidad de servicio implementado cumple con los parámetros establecidos para los diferentes tipos de tráfico, dándole prioridad a tráfico en tiempo real, como de voz, que son más sensibles al retardo y a jitter. Cayendo dentro de los parámetros tolerables: latencia ≤ 150 ms, jitter ≤ 30 ms, y entrega de paquetes de 99.0%.
- El despliegue de calidad de servicio para los entornos IPv6 es muy estable, ya que se utilizan los mismos conceptos de Servicios Diferenciados, que llevan años de depuración.
- La calidad de servicio en IPv6, da lugar a la implementación de un servicio escalable y sin lugar a múltiples interpretaciones.
- Para futuros trabajos, pueden implementarse nuevas metodologías de marcado de tráfico en IPv6 a través de las etiquetas de flujo de su cabecera. Para realizar una comparación entre el desempeño que se tenga con esta clasificación, con relación al utilizado en el actual documento.

ACRÓNIMOS

- **AF**: Assured Forwarding. Reenvío Asegurado.
- **CQ**: Custom Queuing. Encolamiento personalizado.
- **DiffServ**: Differentiated Services. Servicios Diferenciados.
- **DSCP**: Differentiated Services Code Point. Código de servicios diferenciados.
- **EF**: Expedited Forwarding. Renvío Expedito.
- **FIFO**: First In First Out. Primero que entra, primero que sale.
- **FQ**: Fair Queuing. Encolamiento Justo.
- **IETF**: Internet Engineering Task Force.
- **IntServ**: Integrated Services. Servicios Integrados.
- **LLQ**: Low Latency Queuing. Encolamiento de baja latencia.
- **MDRR**: Modified Deficit Round Robin.
- **PBR**: Policy Based Routing. Enrutamiento basado en políticas.
- **PHB**: Per Hop Behavior. Comportamiento por saltos.
- **PQ**: Priority Queuing. Encolamiento de prioridad.
- **QoS**: Quality of Service. Calidad de servicio.
- **RED**: Random Early Detection. Detección Temprana aleatoria.
- **RSVP**: Resource Reservation Protocol. Protocolo de reserva de recursos.
- **RTT**: Round Trip Time. Tiempo de ida y vuelta.
- **SLA**: Service Level Agreement. Acuerdo de nivel de servicios.
- **TC**: Traffic Class. Clase de Tráfico.
- **ToS**: Type of Service. Tipo de servicio.
- **TS**: Traffic Shapping. Modelamiento de tráfico.
- **WRED**: Weighted Random Early Detection. Detección Temprana aleatoria ponderada.

BIBLIOGRAFÍA

[1] Li, Qing. Jinmei, Tatuya, coaut. Shima, Keiichi, coaut. Título: "IPv6 core protocols implementation". Boston: Elsevier, 2007.

[2] Murillo Paternina, Johise. Título: "Protocolo IPv6 y su implementación en las redes de avanzada en Colombia". Cartagena de Indias, 2009. Universidad Tecnológica de Bolívar.

[3] Introducción a IPv6. Disponible en: http://technet.microsoft.com/es-es/library/cc739688(WS.10).aspx. Acceso: 30 de Junio de 2011.

[4] Almeida, J. Intriago, M. Velasteguí, T. Masapanta, I. Mosquera, S. Título: Protocolos de enrutamiento para IPv6. Junio 12 de 2005. 30p.

[5] Álvarez Moraga, Sebastián A.; González Valenzuela, Agustín J. Título: "Estudio y configuración de calidad de servicio para protocolos IPv4 e IPv6 en una red de fibra óptica WDM".

[6] Leinen, Simon. One way delay. http://kb.pert.geant.net/PERTKB/OneWayDelay. Consulta: Diciembre 10, 2011.

[7] Salcedo Parra, Octavio J.; López, Danilo; Ríos, Ángela. Título: "Desempeño de la calidad de servicio sobre IPv6". Artículo de investigación Conciencias. Febrero 1 de 2011.

[8] Alarcón Llamas, Ricardo. Título: "Estudio e implementación de mecanismos de calidad de servicio sobre una arquitectura de servicios diferenciados". Universidad Politécnica de Cartagena. Enero 2003.

[9] IP SLA Configuration Guide, Cisco IOS. Release 15.1MT.

[10] Díaz Cervantes, Lisset. Evaluación de la herramienta GNS3 con conectividad a enrutadores reales. Escuela politécnica superior de Ingeniería de telecomunicación de Barcelona.

[11] Nieto Porras, Luisana B. "Diseño y configuración de calidad de servicio en la tecnología MPLS para un proveedor de servicios de Internet". Escuela politécnica nacional. Quito, Mayo 2010.

[12] Jara Saba, Felipe Ernesto. "Estudio e implementación de una red IPv6 en la UTFSM". Universidad Técnica Federico Santa María. Valparaíso, Chile. Abril de 2009.

[13] Armitage, Grenville. "Quality of Service in IP networks". Editorial: New riders. Primera edición. Abril 07, 2000.

[14] “Implementing Cisco Quality of Service (QOS) v2.0”. Student Guide.

[15] Felici, Santiago. Práctica: Calidad de servicio (CoS y QoS).

[16] http://www.redescisco.net/v2/art/direccionamiento-basico-con-ipv6-ripng/

[17] http://librosnetworking.blogspot.com/2006/11/qu-es-autoqos_10.html

[18] http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/configuration/xe-3s/ipv6-xe-3s-book.pdf

[19] http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/configuration/xe-3s/ip6-qos.html

[20] http://bogpeople.com/networking/dscp.shtml

[21] http://www.gns3.net/gns3-hosts-topologies/

[22]http://www.cisco.com/en/US/docs/ios/12_2t/qos/command/reference/qftcmd8.html

[23]http://docwiki.cisco.com/wiki/IOS_IP_SLAs_for_IPv6_White_Paper

[24]http://www.cisco.com/en/US/tech/tk543/tk757/technologies_tech_note09186a00800949f2.shtml