Revista Científica y Tecnológica UPSE Vol. IV. N°3 Pág. 74 -82 Dic. 2017

8

Evaluación de Parámetros de QoS en una Red VPN-MPLS Diffserv bajo un Entorno Completo de Emulación de Software Libre.

Miroslava Zapata, Franklin Pacheco, Edison De la Torre, María Vallejo.

Recibido: Junio de 2017
Aprobado: Noviembre de 2017

Evaluación de Parámetros de QoS en una Red VPN-MPLS Diffserv bajo un Entorno Completo de Emulación de Software Libre

QoS Parameters Evaluation in a VPN-MPLS Diffserv Network under a Complete Free Software Emulation Environment

Miroslava Zapata, Franklin Pacheco, Edison De la Torre, María Vallejo
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas – ESPE
mazapata@espe.edu.ec

Resumen

El uso de redes de redes privadas virtuales (VPN-MPLS) se ha vuelto muy común dentro de las empresas gracias a sus múltiples ventajas tales como, la comunicación privada a través de una infraestructura de red pública entre sitios geográficamente diversos. Esto lleva a la necesidad de una red eficiente en términos de calidad de servicio (QoS) para garantizar la fiabilidad y la seguridad de la información. Sin embargo, la implementación de una red VPN-MPLS no es fácil ni económica para las pequeñas y medianas empresas; por lo tanto, en la mayoría de los casos, se requiere usar emuladores que tampoco son gratis. La presente investigación analizó una red VPN-MPLS en términos de métricas QoS: delay, jitter y packet loss. Esta evaluación se realizó en un entorno virtual utilizando sólo herramientas de software libre bajo dos escenarios de prueba, con y sin Servicios Diferenciados (DiffServ). Los resultados mostraron que una red VPN-MPLS DiffServ reduce el delay en aproximadamente 96.78% en VoIP, 39.21% en Datos y 66.83% en Streaming; Además, el jitter se redujo en aproximadamente 27,88% en VoIP y 41,09% en datos.

Palabras clave: *QoS, VPN-MPLS, D-ITG, DiffServ, Jitter, Delay*

Abstract

The use of Virtual Private Network – Multi Protocol Label Switching (VPN-MPLS) networks has become very common inside enterprises thanks to their multiple advantages; such as, the private communication across a public network infrastructure between geographically diverse sites. This leads to a need for an efficient network in terms of Quality of Service (QoS) to guarantee reliability and security of information. However, the implementation of a VPN-MPLS network is neither easy nor cheap for small and medium companies; hence, in most cases, it is required the use of emulators that are not free either. In this paper, we analyze a VPN-MPLS network in terms of QoS metrics: delay, jitter and packet loss. This evaluation was performed in a virtual environment using only free software tools under two test scenarios, with and without Differentiated Services (DiffServ). The results showed that a VPN-MPLS DiffServ network reduces the delay by approximately 96.78% in VoIP, 39.21% in Data and 66.83% in Streaming; furthermore, the jitter was reduced by approximately 27.88% in VoIP and 41.09% in Data.

Keywords: *QoS, VPN-MPLS, D-ITG, DiffServ, Jitter, Delay*

1. Introducción

En el mundo de las comunicaciones, los requerimientos de los usuarios finales y clientes empresariales son cada vez más exigentes en cuanto a concentración de servicios, velocidad de transmisión y QoS (del inglés *Quatlity of Service*) en los distintos tipos de tráfico [1]. Varios avances tecnológicos a nivel de red de acceso y core se han desarrollado, siendo uno de los más importantes las redes VPN-MPLS (del inglés *Virtual Private Network Multi Protocol Label Switching*). Estas redes permiten garantizar la información entre diferentes sitios remotos de una organización, a través de una red pública como internet, con una mayor flexibilidad, eficiencia, seguridad y bajo costo, sin la necesidad de utilizar comunicaciones dedicadas punto a punto [1]. Adicionalmente, uno de los principales requerimientos en las redes corporativas actuales es proveer QoS mediante diferentes mecanismos. Sin embargo, para redes de gran extensión como internet, es recomendable utilizar DiffServ (del inglés *Differentiated Services*) que permite resolver el problema de escalabilidad no soportado por otros mecanismos como IntServ (del inglés *Integrated Services*) [2].

Los nuevos requerimientos y desarrollos tecnológicos han creado una tendencia a nivel mundial en migrar las antiguas tecnologías como ATM (del inglés *Asynchronous Transfer Mode*) y Frame Relay hacia redes de nueva generación VPN-MPLS con QoS [3]. Sin embargo, el costo de implementar una red con estas características resulta muy significativo para las pequeñas y medianas organizaciones, quienes previo a adquirir el equipamiento necesario, deben realizar una emulación de la misma y determinar las ventajas que obtendrían con la implementación de la nueva infraestructura. Estas emulaciones han sido abordadas a través de herramientas de software licenciado que también implican un costo. Sin embargo, hasta el momento no se han emulado en un entorno completo de software libre que implique cero costo y facilidad de experimentación [4].

El impacto de QoS en redes VPN-MPLS ha sido analizado en otros trabajos relacionados en los que se han utilizado equipos reales en entornos de laboratorio o emulaciones con herramientas de software licenciado. Es así, que en [5] se muestra una propuesta de arquitectura MPLS/DiffServ para proveer mecanismos de QoS en el transporte de la telefonía IP (del inglés *Internet Protocol*), en la que se calculan las rutas menos congestionadas a través de una subred, donde el tráfico de voz es separado del de datos para resolver el problema de flujo multiproducto de los paquetes que transportan la voz. La provisión de QoS mediante la integración de MPLS y Diffserv fue propuesta en [6] y en [7], en los cuáles se realizó una evaluación de QoS en redes Wimax basadas en IP/MPLS; se utilizó el mecanismo DiffServ y se compara el envío de flujo de datos susceptibles a retardos en las tecnologías MPLS, MPLS-TP (del inglés *Multiprotocol Label Switching - Transport Profile*) y GMPLS (del inglés *Generalized Multi-Protocol Label Switching*). Finalmente, en [8] se aplicó la arquitectura Diffserv sobre redes MPLS para la provisión de QoS punto a punto en la transmisión de tráfico en tiempo real.

En la presente investigación se evaluaron los parámetros de QoS para tres escenarios: IP, MPLS y MPSL con Diffserv, determinándose que la red MPLS con Diffserv resultó ser la más adecuada.

En la literatura no se ha encontrado hasta el momento emulaciones de este tipo de redes bajo un entorno virtual de software libre, es así, que el objetivo de este trabajo fue evaluar los parámetros de QoS *jitter* (variación del retardo), *delay* (retardo) y *packet loss* (pérdida de paquetes) en una red VPN-MPLS con y sin DiffServ, emulada bajo un entorno completamente virtual, con la utilización de herramientas de software libre para configurar, manipular y analizar los diferentes tipos de tráfico. El reto radica en el diseño de la red VPN-MPLS, la selección de las herramientas adecuadas de software libre, la configuración de la red, la evaluación de los parámetros de QoS con y sin el mecanismo de Diffserv.

El presente artículo está organizado de tal manera que, en la sección 2 se presenta el marco conceptual relacionado al presente trabajo; en la sección 3 se detalla el diseño y emulación de la red VPN-MPLS en el entorno virtual; en la sección 4 se presenta la evaluación de los parámetros de QoS y los resultados obtenidos en comparación con los estándares de la UIT-T G.1010, Y.1541, IEEE 8021.1p resumidos en [8]. Finalmente, en la sección 5 se presentan las conclusiones de la investigación.

2. Marco Conceptual

2.1. Arquitectura VPN-MPLS

Las Redes VPN-MPLS son redes eficientes, fiables y escalables, utilizadas para la interconexión de las distintas redes de una empresa espaciada geográficamente. El resultado es una plataforma WAN (del inglés *Wide Area Network*) completamente gestionada y con un bajo costo [9].

2.2. Calidad de Servicio (QoS)

Al hablar de QoS, se puede abordar desde dos puntos de vista: En primer lugar, para el usuario, QoS es la percepción de que el servicio funciona adecuadamente,

por ejemplo, en una conversación telefónica la comunicación no debe entrecortarse. En segundo lugar, para el personal responsable de la red, QoS es la posibilidad de maximizar el ancho de banda sin degradar las aplicaciones que se encuentran en ejecución. Para este propósito se debe considerar parámetros que no degraden una transmisión en tiempo real, como el control del *delay*, *jitter* y *packet loss* [10].

2.3. Mecanismo DiffServ

El mecanismo DiffServ consiste en clasificar el tráfico para ofrecer distintos niveles de QoS de acuerdo a las necesidades de los clientes facilitando estabilidad y despliegue en las redes [11].

2.4. Parámetros de QoS

- **Delay***:* También llamado latencia o retardo, es la demora o tiempo que tarda un paquete en ser transferido de un origen a un destino [8].
- **Jitter***:* Debido al retardo que se produce en los flujos de datos se genera un problema llamado *jitter* que aparece por congestión en la red, especialmente por una incorrecta sincronización de bits entre sus elementos [8].
- **Packet Loss:** Es el descarte de paquetes que no llegan a tiempo al receptor [8].

3. Diseño de la red

La red VPN-MPLS diseñada en el presente trabajo está constituida por una red matriz y una red sucursal interconectadas mediante una red MPLS como se muestra en la Figura. 1. Se utilizan diferentes protocolos de enrutamiento en las distintas redes. En la red MPLS se utiliza el protocolo OSPF (del inglés *Open Shortest Path First*) para mejorar su rendimiento, en la red matriz se emplea el protocolo EIGRP (del inglés *Enhanced Interior Gateway Routing Protocol*) para acelerar la convergencia en la red e intercambiar información, y en la red sucursal se utiliza el protocolo OSPF para permitir escalabilidad y estabilidad en la red. Para enlazar EIGRP de la red matriz con OSPF de la red sucursal se utiliza el protocolo BGP (del inglés *Border Gateway Protocol*) que permite la redistribución de rutas. Adicionalmente, se crean redes privadas virtuales para proveer seguridad con VRF (del inglés *Virtual routing and forwarding*) que actúa como un router lógico que permite definir caminos virtuales para diferentes clientes.

Para el diseño de la red se contemplan las siguientes etapas:

a) Requerimientos.
b) Selección de la topología de la red
c) Asignación de direccionamiento y equipo
d) Emulación de la red
e) Calidad de Servicio en la red

3.1. Requerimientos

Se definieron dos tipos de requerimientos: de usuario y de servicio. Dentro de los requerimientos de usuario se toma en cuenta el tiempo de respuesta, confiabilidad, adaptabilidad y políticas de seguridad de los equipos. Dentro de los requerimientos de servicio se consideran los servicios de videoconferencia, navegación Web, correo electrónico y servicio de voz.

De los requerimientos indicados anteriormente, en el presente trabajo se emplean los siguientes tipos de tráfico: voz, datos críticos, datos de administración, datos generales y *Best Effort* (Mecanismo de mejor esfuerzo).

3.2. Selección de la Topología de la Red

3.2.1. Topología de la Backbone

Una vez analizados los requerimientos, se define utilizar la topología de red *Full Mesh* (Malla completa) que conecta cada nodo con el resto de nodos para crear redundancia y resistencia a fallas. Esta topología es la más adecuada por conectividad de redundancia que es característica de las VPN-MPLS. Adicionalmente, se busca que cada ruta en el núcleo tenga el *next-hop* (Próximo salto) más cercano hacia el destino, para tener alta disponibilidad, caminos redundantes y tolerancia a las fallas entre los equipos.

3.2.2. Topología de la red Matriz y de la red Sucursal

La topología que se utiliza tanto en la red matriz como en la red sucursal es la configuración estrella debido a que es escalable, fácil de configurar y de mantenimiento económico.

3.3. Asignación de direccionamiento y equipo

Se trabajó con direccionamiento privado y subneteo con máscara /30 para el backbone de la MPLS, los ruteadores de frontera del cliente CE (del inglés *Customer Edge router*), los routers del cliente de la red matriz y de la red sucursal, lo cual permite la optimización del espacio de direccionamiento. Para los *host* de los ruteadores cliente se trabajó con un direccionamiento privado clase B para un mayor crecimiento de usuarios tanto para la red matriz como para la red sucursal.

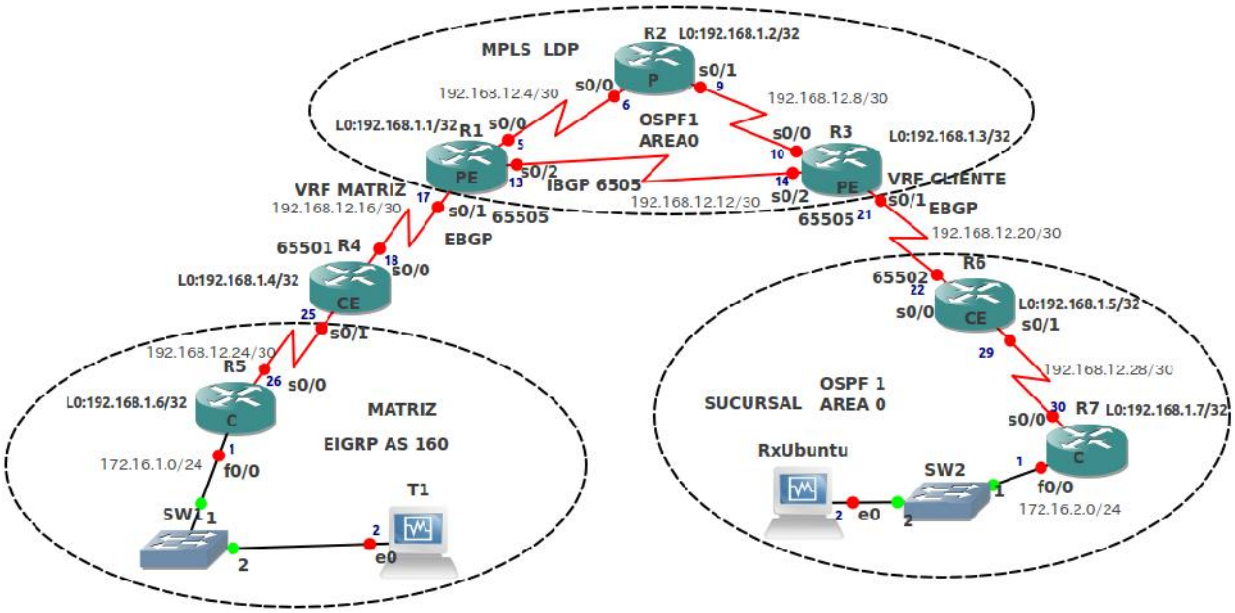

Figura 1. Topología completa de la red VPN-MPLS

En lo que respecta a los equipos simulados para el *backbone* se utiliza el ruteador Cisco 3745 con IOS versión mínima 12.4 T, que tiene características VPN-MPLS, el mismo que se encarga de recibir todo el tráfico que le llega y transferirlo a los dispositivos PE *(*del inglés *Provider Edge*) (Cisco 3745) donde se concentra toda la información para enviarse a la red matriz y a la red sucursal.

3.3. Emulación de la Red

La emulación del proyecto se la realizó con el software especializado GNS3 [12], una herramienta en tiempo real para realizar topologías complejas de red. Adicionalmente, permite la virtualización de sistemas operativos mediante herramientas como VirtualBox [13] y el software D-ITG (del inglés *Distributed Internet Traffic Generator*) [14]; para inyectar tráfico y analizar los parámetros de QoS. A través de D-ITG se emplea una técnica intrusiva de inyección de tráfico que permite evaluar los parámetros de QoS de manera diferente a la que se realiza empleando equipos reales.

3.4. Calidad de Servicio en la Red

En calidad de servicio se realizan las siguientes consideraciones:

1. Tipos de tráfico
2. Asignación de ancho de banda
3. Mecanismo de QoS DiffServ

3.5.1. Tipos de tráfico

Para este trabajo se utilizan los tipos de tráfico recomendados por Cisco Press [15], para un entorno empresarial: voz, datos críticos empresariales (datos transaccionales), datos de administración (administración de la red como IP *routing*), datos generales (*ping*, *traceroute*, entre otros) y *Best effort* (*Scavenger*).

3.5.2. Asignación de Ancho de Banda

En la red de prueba se considera un ancho de banda total de 64 Kbps, a cada tipo de tráfico se le asigna un porcentaje del ancho de banda total en función de las recomendaciones dadas por Cisco Press [15]. En la Tabla 1 se detalla la asignación para los diferentes tipos de tráfico. Se asigna el 15 % para voz para evitar que se degrade la conversación; el 20 % para datos críticos empresariales para asegurar el envío correcto de la información; el 10 % para datos de administración para protocolos de enrutamiento; el 10 % para datos generales de la red (*ping*); y el 5 % para *Best Effort* donde están los *scavenger*, que son un tipo de tráfico que perjudica a la empresa; por ejemplo, videos, video juegos, redes sociales, entre otros.

3.5.3 Mecanismo de calidad de servicio Diffserv

Para utilizar el mecanismo *DiffServ* se debe clasificar el servicio y posteriormente el marcaje de acuerdo al tipo de tráfico.

Tabla 1.- Asignación de ancho de banda

Tipo de tráfico	Asignación de AB
Voz	15%
Datos críticos empresariales	20%
Datos de Administración	10%
Datos Generales	10%
Best effort (Scavenger)	5%

3.5.3.1. Marcaje de los paquetes

Se realizó la programación de las clases con los valores de *IP Precedence* y DSCP (del inglés *Differentiated Services Code Point*) para realizar el marcaje de los paquetes. Con *IP Precedence* se puede tener 8 diferentes marcaciones, mientras que con DSCP se logra mayor granularidad, pues se obtienen 64 marcaciones. En el presente trabajo se usa la combinación de DSCP e *IP Precedence*, mostrado en la Tabla 2. Como se puede observar, el tráfico de voz tiene la mayor prioridad, marcándolo como DSCP EF (código de marcaje) que está asignada en la RFC 2598/3246 [16], [17] para que el usuario final pueda recibir adecuadamente la transmisión y no se degrade la conversación.

Tabla 2. Marcaje de Paquetes

Tipo de tráfico	Clase de Servicio	Código de Marcaje
Voz	EF	DSCP EF
Datos críticos empresariales	Clase 4	Precedence 4
Datos de Administración	Clase 3	Precedence 3
Datos Generales	Clase 2	Precedence 2
Best effort (Scavenger)	Clase 1	Precedence 1

Para la marcación con *IP Precedence* se tiene: *Precedence 4* para datos críticos , *Precedence 3* para datos de administración donde están los protocolos de enrutamiento, *Precedence 2* para datos generales como *ping*, y *Precedence 1* para *Scavenger* también llamado *Best Effort*.

3.5.3.2. Ubicación del marcaje en la red

Existen dos posibles escenarios para ubicar el marcaje, ya sea en el lado del cliente o en el del proveedor. Si el cliente impone su marca, el ancho de banda contratado no permitirá pérdida de paquetes y no habrá reenvió de los mismos; por lo tanto el costo para el cliente es alto. Por otro lado, si el proveedor impone su marca, no hay aseguramiento de la calidad de servicio al cien por ciento para el cliente, porque el proveedor tiene el control del tráfico que ingresa a la red VPN-MPLS; por lo tanto, el servicio que cobra es más barato. Sin embargo, cuando se excede el tráfico contratado se descartan los paquetes, se incrementa el *delay* y también los reenvíos de paquetes en la red convirtiéndose en un grave problema para el cliente.

Debido a que es preferible reducir al mínimo la pérdida de paquetes, la marcación del tráfico debe hacerse lo más cerca posible al cliente, debido a que ciertos tipos de tráfico como voz y video deben ser remarcados antes de pasar al proveedor de servicios. Adicionalmente, los servicios que ofrecen los proveedores evolucionan o se expanden a través del tiempo y una manera fácil de ajustarse a estos cambios es el remarcado en el borde del *Customer Edge* (CE). Por lo tanto, en el presente trabajo, el marcaje se realiza en el borde de salida del CE y no dentro de la red matriz.

4. Evaluación de la Calidad de Servicio

En esta sección se analizan los parámetros de calidad de servicio: *delay*, *jitter* y *packet loss*. En primer lugar se revisan las recomendaciones de la UIT-T G.1010, Y 1541 y la IEEE 8021.1p y posteriormente se comparan con los valores obtenidos mediante la utilización del mecanismo QoS *DiffServ*.

4.1. Parámetros de Acuerdo a los Estándares

Los valores establecidos en las recomendaciones de la UIT-T G.1010, Y 1541 y la IEEE 8021.1p se sintetizan y clasifican cualitativamente en [8] como se indican a continuación:

4.1.1. Delay

Como se muestra en la Tabla 3 [8], el *delay* máximo permitido para VoIP es de 150ms con el fin de que no se solape la conversación, 300 ms para Datos para que exista una adecuada transmisión, y 250 ms para *streaming* para que evitar la degradación del video.

Tabla 3. Rangos para valorar Delay cualitativamente

Tráfico	Excelente	Muy Bueno	No Adecuado
VoIP	<100ms	>100ms y <150ms	>150ms
Datos	< 250ms	>250ms y <300ms	>300ms
Streaming	<=100 ms	>100ms y <=250ms	> 250ms

4.1.2. Jitter

Como se muestra en la Tabla 4 [8], el *jitter* máximo para que no haya una variación de flujo de datos entre el transmisor y el receptor es 50 ms para VoIP, 70 ms para datos y 65 ms para *streaming*.

Tabla 4. Rangos para valorar Jitter cualitativamente

Tráfico	Excelente	Muy Bueno	No Adecuado
VoIP	<40ms	>40ms y <50ms	>50ms
Datos	< 55ms	>55ms y <70ms	>70ms
Streaming	<=35 ms	>35ms y <=65ms	> 65ms

4.1.3. Packet loss

Se establece un valor máximo del 3 % para VoIP, 5 % para Datos y 5 % para *Streaming* como se observa en la Tabla 5 [8].

Tabla 5. Rangos para valorar packet loss cualitativamente

Tráfico	Excelente	Muy Bueno	No Adecuado
VoIP	<1%	>1% y <3%	>3%
Datos	< 3%	>3% y <5%	>5%
Streaming	<=2%	>2% y <=5%	> 5%

4.2. Instrumentos de Análisis de Tráfico

Se utilizan los siguientes instrumentos de medición para poder evaluar los parámetros de QoS en la red de análisis: Wireshark y D-ITG [18].

Wireshark es un *sniffer* que permite capturar, filtrar y analizar tráfico de la red. Permite analizar detalladamente los paquetes y mostrar los resultados en un entorno gráfico [18].

D-ITG es un software para inyectar tráfico y analizar los parámetros de calidad de servicio como *delay*, *jitter* y *packet loss*. En el presente trabajo se inyecta tráfico real de datos, *Streamming* y VoIP con los códec G.711, G.729 y G.723.1 [14]. Se utilizan dos máquinas virtuales, la una en la red matriz y la otra en la red sucursal. En cada máquina virtual se instala Ubuntu como sistema operativo y el inyector de tráfico D-ITG en un *host* como transmisor y en el otro *host* como receptor.

4.3. Mediciones y resultados

Los parámetros de calidad de servicio como *delay*, *jitter* y *packet loss* fueron medidos con los instrumentos indicados en la sección anterior. Para el tráfico de Voz y *Streaming* se consideró el protocolo UDP (del inglés *User Datagram Protocol*) y para el tráfico de datos se consideró el protocolo TCP (del inglés, *Transmission Control Protocol*) para que los datos lleguen completos. Las mediciones se realizaron sin configurar QoS y luego con QoS mediante *DiffServ*.

4.3.1. Delay

Los valores de *delay* medidos se muestran en la Figura 2. Como se puede observar, existe una gran diferencia entre los valores obtenidos sin *DiffServ* y con *DiffServ*. Por ejemplo, para el tráfico de VoIP con códec G.711 el *delay* promedio cambió de 719 ms sin *DiffServ* a 23 ms con *DiffServ*, es decir que la reducción fue del 96.80 %; por otro lado, para *Streaming* la reducción fue del 66.83 %. Los porcentajes de reducción en *delay* de todos los tipos de tráfico pueden observarse en la Tabla 6. Adicionalmente, al realizarse la valoración cualitativa de *delay* utilizando la Tabla 3, se puede observar que sin utilizar *DiffServ* no se obtienen valores adecuados para la correcta operación de la red VPN-MPLS, mientras que al utilizar *DiffServ* los valores se encuentran en el rango recomendado por los estándares para proporcionar QoS. La valoración cualitativa de *delay* para todos los tipos de tráfico se observa en la Tabla 6.

4.3.2. Jitter

Los valores de *jitter* medidos se muestran en la Figura 3, Tabla 7. Como se puede observar, los valores de *jitter* sin *DiffServ* y con *DiffServ* también presentan diferencias considerables como ocurre con *delay*. Por ejemplo, para el tráfico de VoIP con códec G.711 el *jitter* promedio disminuyó de 1 ms a 0.68 ms, es decir que la reducción fue del 32 %. Sin embargo, para *Streaming* mas bien existió un incremento del 7.53 %. En realidad los valores medidos de *jitter* sin *DiffServ* están dentro de los rangos de operación recomendados por los estándares de acuerdo al Tabla 4; sin embargo, se obtienen mejoras al aplicar QoS con el mecanismo *DiffServ* como se observa en la Tabla 8.

Tabla 6. Comparación cualitativa de delay

Tráfico	G.711	G.723.1	G.729	Datos	Streaming
VPN-MPLS	No adecuado	No adecuado	Muy Bueno	Muy Bueno	Muy Bueno
VPN-MPLS y DiffServ	Muy Bueno	Excelente	Excelente	Excelente	Excelente

Tabla 7. Comparación cualitativa de *jitter*

Tráfico	G.711	G.723.1	G.729	Datos	Streaming
VPN-MPLS	Muy Bueno	Muy Bueno	Muy Bueno	Muy Bueno	Muy Bueno
VPN-MPLS y DiffServ	Excelente	Excelente	Excelente	Excelente	Excelente

Tabla 8. Porcentaje de reducción de parámetros de QoS al utilizar DiffServ

Tráfico	G.711	G.723.1	G.729	Datos	Streaming
Delay	96.80%	96.96%	96.58%	39.21%	66.83%
Jitter	32.00%	30.28%	21.36%	41.09%	-7.53%

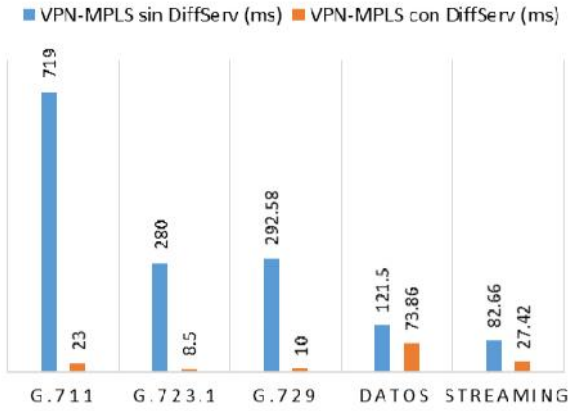

Figura 2. Valores medidos de *delay*

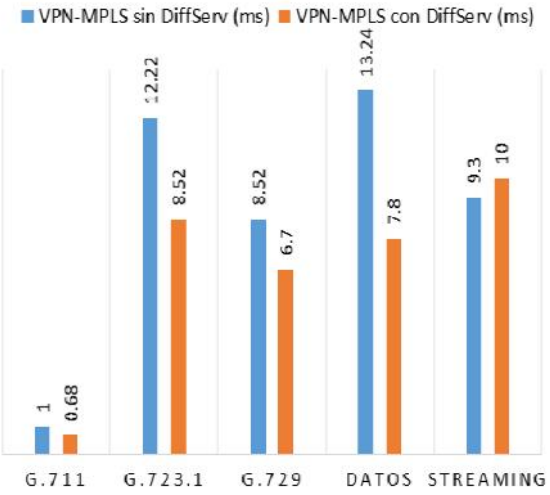

Figura 3. Valores medidos de jitter

4.3.3. Packet Loss

Tanto en las mediciones realizadas sin QoS y con QoS *DiffServ* no hubo pérdida de paquetes porque en redes VPN - MPLS los paquetes se clasifican, se marcan y se procesan de manera eficiente.

5. Conclusiones

Se diseñó y emuló una red VPN-MPLS con y sin QoS a través de *DiffServ* en un entorno virtual únicamente con la utilización de herramientas de software libre. Los resultados obtenidos pueden ser replicados por cualquier organización, sin que le represente costo alguno, para evaluar el desempeño que tendrían en sus redes, previo a la adquisición de equipos.

Con los resultados de *delay* cuantitativa y cualitativamente, se observa que al utilizar *DiffServ* se logra una reducción promedio del 96.78 % en voz, 39.21 % en datos y 66.83 % en *Streaming*, lo cual muestra que *DiffServ* es capaz de ofrecer un excelente *delay* en todos los tipos de tráfico.

Los resultados de *jitter* cuantitativa y cualitativamente demostraron mejoras significativas al emplear *DiffServ*. Se logró una reducción promedio del 27.88 % en voz, 41.09 % en datos y -7.53% en *Streaming*, demostrando que *DiffServ* puede ofrecer un excelente *jitter* en todos los tipos de tráfico. A pesar de que *DiffServ* incrementó el *jitter* en *Streaming*, se pueden hablar de valores excelentes para una red VPN-MPLS.

Para ejecutar la emulación presentada en este trabajo, se debe disponer de máquinas con capacidad de procesamiento y memoria suficientes para usar las herramientas GNS3 y D-ITG (mínimo 4GB de memoria RAM y 4GHz de velocidad de procesamiento de la PC) que permitan simular plataformas robustas como Cisco y analizar el tráfico de la red. Además se debe configurar adecuadamente el protocolo de la capa de transporte TCP o UDP, ya que de estas configuraciones dependen los resultados de la evaluación de la calidad de servicio.

6. Referencias Bibliográficas

[1] R. Damian, “Transmisión de voz, video y datos en Redes Privadas Virtuales VPN/MPLS.,” Universidad de Belgrano-Facultad de Ingeniería y Tecnología Informática-Licenciatura en Sistemas, 2008.

[2] E. Mykoniati *et al.*, “Admission control for providing QoS in DiffServ IP networks: the TEQUILA approach,” *IEEE Communications Magazine*, vol. 41, no. 1, pp. 38–44, Jan. 2003.

[3] M. Huerta, X. Hesselbach, and O. Calderon, “Problemas abiertos en MPLS. Migración, Protección, Gestión de Recursos y Balanceo de Carga,” presented at the III Workshop MPLS networks, Girona-España, 2004, vol. 3.

[4] Y. Jia, M. L. Guerrero, O. Kabranov, D. Makrakis, and L. O. Barbosa, “Design and testbed implementation of adaptive MPLS-DiffServ enabled virtual private networks,” in *CCECE 2003 - Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No.03CH37436)*, 2003, vol. 2, pp. 965–968 vol.2.

[5] S. de Oliveira Guerra, “Una propuesta de arquitectura MPLS/DiffServ para proveer mecanismos de calidad de servicio (QOS) en el transporte de la telefonía IP,” phd, E.T.S.I. Telecomunicación (UPM), 2004.

[6] R. Jiménez Mateo, C. Paniagua, A. Gazo Cervero, J. González Sánchez, and F. Rodríguez Pérez, “Integración de MPLS y DiffServ en una Arquitectura para la Provisión de QoSc.” 2004.

[7] R. C. Garcia, B. S. Reyes Daza, and O. J. Salcedo, “Evaluation of Quality Service Voice over

Internet Protocol in WiMAX Networks based on IP/MPLS Environment," 2015, pp. 59–66.
[8] P. A. Buñay Guisñan, "Aplicación de la arquitectura Deffserv sobre redes MPLS para la provisión de QoS punto a punto en la transmisión de tráfico en tiempo real," Mar. 2013.
[9] O. Piña and G. Diana, "Diseño y comparación de redes de acceso MPLS y Metro Ethernet integradas a un backbone MPLS para un proveedor de servicios y realización de un prototipo base," Thesis, Quito, 2016., 2016.
[10] M. Zapata Rodríguez, "Evaluación de parámetros de calidad de servicio (QOS) para el diseño de una red VPN con MPLS.," 2016.
[11] T. García Reyes, "Análisis de los modelos de servicios diferenciales y servicios integrales para brindar QoS en Internet," Universidad Tecnológica de la Mixteca, Oaxaca-México, 2007.
[12] GNS3 Technologies Inc, "GNS3 The software that empowers network professionals," 2017. [Online]. Available: https://www.gns3.com/. [Accessed: 26-May-2017].
[13] Oracle, "VM VirtualBox," 2017. [Online]. Available: https://www.virtualbox.org/. [Accessed: 26-May-2017].
[14] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, Oct. 2012.
[15] J. Guichard and I. Pepelnjak, "MPLS/VPN Architecture Overview." Cisco Press, 2002.
[16] V. Firoiu, A. Charny, and B. Davie, "An Expedited Forwarding PHB (Per-Hop Behavior)." [Online]. Available: https://tools.ietf.org/html/rfc3246. [Accessed: 26-May-2017].
[17] K. Nichols, V. Jacobson, and K. Poduri, "An Expedited Forwarding PHB." [Online]. Available: https://tools.ietf.org/html/rfc2598. [Accessed: 26-May-2017].
[18] G. Combs, "Wireshark," 2017. [Online]. Available: https://www.wireshark.org/. [Accessed: 26-May-2017].