

# Propuesta para gestión de QoE en SDN

## SDN QoE management proposal

César Alonso Irizar<sup>1</sup>, Caridad Anías Calderon<sup>1</sup>

<sup>1</sup>Universidad Tecnológica de La Habana, José Antonio Echeverría, Cujae, La Habana, Cuba

Correo electrónico: cesar.860709@gmail.com

Este documento posee una licencia Creative Commons Reconocimiento/No Comercial 4.0 Internacional 

Recibido: 5 de julio de 2018    Aprobado: 23 de noviembre de 2018

### Resumen

Las redes definidas por software han generado grandes expectativas en los últimos años, debido a que poseen una arquitectura dinámica, rentable y adaptable para dar respuesta a los requerimientos de las redes de telecomunicaciones y a los proveedores de servicios. Nuevas características, funciones y servicios de red pueden ser implementados en forma de aplicaciones, determinando de esta manera el comportamiento de la red. Con el objetivo de facilitar y acelerar el proceso de desarrollo de aplicaciones, han surgido nuevos lenguajes para la programación SDN, los cuales brindan las herramientas para extender los beneficios prometidos por este paradigma. En el presente trabajo se aborda el desarrollo actual de las SDN, prestando especial atención a la gestión de QoE en las mismas. Para esto se elabora un procedimiento que permite evaluar y gestionar la QoE basado en parámetros de QoS y tomar acciones para mejorarla en caso que sea necesario. El procedimiento es validado mediante la creación de una aplicación que emplea el lenguaje de alto nivel de abstracción *frenetic* para la programación de SDN.

Palabras claves: QoE, QoS, VoIP, SDN, procedimiento, método

### Abstract

The Software Defined Networks have generated big expectations in the last years, because they possess a dynamic, profitable and adaptive architecture to give answer to the requirements of the telecommunications networks and service providers. New characteristics, functions and network services can be implemented in form of applications, determining this way the behavior of the network. With the objective of facilitate and accelerate the process of application development, new languages have arisen for the SDN programming, which offer the tools to extend the benefits promised by this paradigm. In this work is approached the current development of SDN, paying special attention to the QoE management. For this a procedure is elaborated, permitting QoE based QoS parameters evaluation and management, and actions to do in order to become it better. Procedure is validated through an application that uses *frenetic* as a high level abstraction language for SDN programming.

Key words: QoE, QoS, VoIP, SDN, procedure, method

## INTRODUCCIÓN

Para lograr un servicio de calidad y minimizar el impacto negativo que puede ocasionar un gran tráfico en la red es muy importante la gestión de los dispositivos que intervienen en dicho servicio. Pero hoy en día no solo es necesario mantener parámetros de Calidad de Servicio (*Quality of Service*, QoS por sus siglas en inglés) en la red, sino que comienzan a requerirse otros criterios que tienen en cuenta cómo el cliente experimenta esa calidad que se evalúa. La evaluación de la Calidad de Experiencia (*Quality of Experience*, QoE por sus siglas en inglés) en servicios IP permite obtener múltiples beneficios tanto a los clientes como a los proveedores de servicios. Debido a esto, el concepto de QoE ha ido tomando cada vez más importancia, ya que permite evaluar el servicio que consumen los usuarios.

Por otra parte, la necesidad de darle robustez a las redes, flexibilidad y facilidad de administración ha conducido al desarrollo de un nuevo paradigma llamado Redes Definidas por Software (*Software Defined Networks*, SDN por sus siglas en inglés), el cual abre las puertas a la programabilidad y a las innovaciones de forma rápida y sencilla.

En las SDN, nuevas características y servicios pueden ser añadidos dinámicamente mediante la programación de aplicaciones que permitan, como en las redes clásicas, realizar todas las funciones inherentes a estas como monitoreo y control, aprovechando el principio de operación de las mismas, para de esta forma mantener en valores elevados la experiencia de usuario, sin embargo, al ser una tecnología novedosa, contar con una guía podría resultar muy favorable para el desarrollo futuro de las SDN.

## MATERIALES Y MÉTODOS

Varios organismos internacionales [1] han definido la experiencia del usuario, como también se le conoce a la QoE; por ejemplo, la Unión Internacional de Telecomunicaciones (UIT) define la calidad de experiencia como: "la aceptabilidad global de una aplicación o un servicio, percibida subjetivamente por el usuario final" [2].

El trabajo desarrollado en [3] propone un algoritmo de encaminamiento basado en Dijkstra para transmisiones de vídeo utilizando OpenFlow y controlador Floodlight que mejora la QoE al usuario.

Otros trabajos de investigación sobre transmisiones de vídeo, por ejemplo [4], propone agregar funcionalidades extra al conmutador OpenFlow. Estas capacidades son: filtros Bloom con reenvío, *Greedy routing* y codificación de la información en la red.

Los autores de [5] proponen un Marco Justo de QoE (*QoE Fairness Framework*, QFF por sus siglas en inglés) con OpenFlow que intenta maximizar la calidad de experiencia del usuario. QFF monitoriza el estado de todos los conmutadores y aplicaciones de vídeo en una red y distribuye dinámicamente los recursos de la red a cada dispositivo.

Por otra parte [6] plantea una arquitectura, donde en la capa de aplicación SDN, se introduce un Servicio de Gestión de Recursos, que se comunica con el controlador SDN. En este modelo se optó por REST API (*Representational State Transfer Application Program Interface*). En la capa de control se implementan dos módulos, uno para la monitorización de la red y otro para el cálculo de rutas óptimas que hagan más eficiente la transmisión del contenido multimedia. Finalmente, en los nodos origen y destino de la comunicación se implementan componentes que interactúen con el software multimedia y sean capaces de soportar el protocolo PCMS (Protocolo de Control Multimedia en SDN), definido para el intercambio de mensajes entre los Agentes de Gestión de Recursos y el Servicio de Gestión de Recursos.

Aunque estos trabajos abordan de diversas formas el tema de este artículo, no se brinda una guía que sea aplicable de forma general a la gestión de la QoE, y además no se tiene acceso al código fuente de las propuestas realizadas. La figura 1 muestra las fases que componen la propuesta para gestionar la QoE en redes SDN.

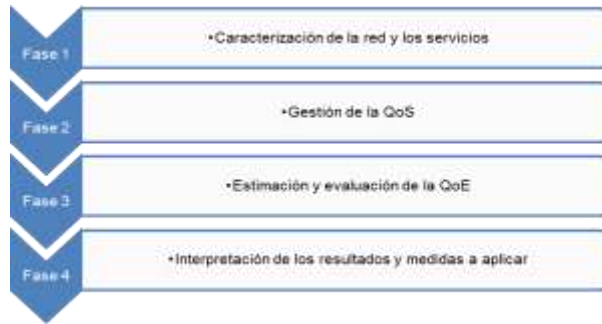


Fig. 1. Procedimiento para gestión de la QoE en SDN

**Caracterización de la red y los servicios**

Para iniciar esta fase (figura 2) se hace necesario obtener, primeramente, información básica sobre el escenario y servicio en cuestión, por ejemplo, interfaces, direcciones MAC e IP involucradas, así como números de puertos. Todo esto con el objetivo de poder monitorear y poder tomar acciones a nivel de SDN. A continuación, se debe recolectar información sobre parámetros relacionados con el servicio propiamente como por ejemplo: codificación, resolución, velocidad de muestreo, velocidad de trama y sincronización entre otros. Esto es importante debido a que puede realizarse un análisis basado en dichos parámetros y cambiarlos de acuerdo con los objetivos que se persigan. Por ejemplo, en VoIP puede ser que para mejorar la QoS se deba cambiar de un CODEC a otro que emplee menor ancho de banda.



Fig. 2. Caracterización de la red y los servicios

**Gestión de la QoS**

Una vez seleccionado el servicio que se va a utilizar para evaluar la QoE se propone seleccionar y obtener los valores de los parámetros de QoS para su posterior evaluación (figura 3).



Fig. 3. Gestión de la QoS

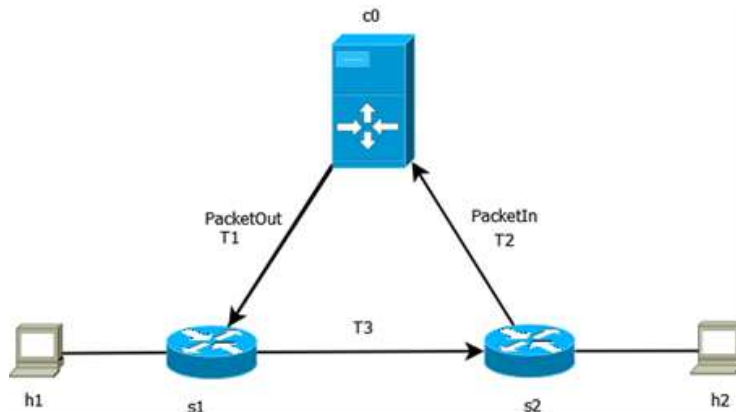
Es muy importante la selección de parámetros de QoS pues existen parámetros de calidad generales como la demora, el *jitter* y la pérdida de paquetes, y otros que son intrínsecos al servicio como, por ejemplo, la tasa de llamadas exitosas en el caso de VoIP, siendo necesario, antes de obtener los valores de dichos parámetros mediante cualquier método o herramienta, realizar la elección de cuáles serán sometidos a evaluación.

Para la gestión de QoS en SDN existen métodos como el enrutamiento de flujos multimedia [7], enrutamiento interdominios [8], reservación de recursos [9], gestión de colas [10] y gestión de políticas QoS [9].

La solución que se propone en este trabajo monitorea la latencia aprovechando el principio de operación de las SDN. En otras palabras, la información sobre un camino se captura directamente de dispositivos de red. La idea principal es usar los mensajes de OpenFlow para medir la demora entre los *switches*.

La siguiente propuesta es general y aplicable a cualquier escenario de red, solo debe tenerse en cuenta el método propuesto y aplicarlo de acuerdo con condiciones existentes.

El primer paso consiste en crear un paquete que será usado como prueba [11]. Para esto el controlador, con un mensaje de tipo PacketOut, pide a un *switch* enviar el paquete, a través de un puerto definido, al próximo *switch* y se registra el tiempo de envío. Finalmente, cuando el próximo *switch* recibe el paquete, envía un mensaje de tipo PacketIn al controlador para comunicarle el estado del paquete y se registra el tiempo de recepción. La figura 4 ilustra el mecanismo descrito.



**Fig. 4. Cálculo de la latencia**

La demora que se quiere obtener corresponde a T3, que es el tiempo que demora el paquete en llegar de un *switch* a otro. Cuando el controlador recibe el mensaje de tipo PacketIn, el tiempo total puede ser hallado como:

$$T_{total} = T1 + T2 + T3 \quad (1)$$

donde:

T1: tiempo que demora el paquete en llegar desde el controlador al *switch* s1.

T2: tiempo que demora el paquete en llegar desde el *switch* s2 al controlador.

La ecuación (1) es equivalente a sustraer del tiempo de recepción del mensaje PacketIn, el tiempo de envío del mensaje de tipo PacketOut

Para calcular tanto T1 como T2 se tiene en cuenta el tiempo de ida y vuelta (*Round Trip Time*, RTT por sus siglas en inglés) que resulta de enviar una pregunta y recibir una respuesta del controlador al *switch* y viceversa. De acuerdo con (2)

$$T1 = 0.5 * (Tb - Ta) \quad (2)$$

Ta es el instante en que se envía una solicitud OpenFlow de tipo ports-stats-request al *switch* 1 para el caso de T1, y Tb es el instante en que se recibe la respuesta al mensaje que es de tipo port-stats-received. El mismo método puede aplicarse para hallar T2. En consecuencia y despejando T3 de (1):

$$T3 = T_{total} - T1 - T2 \quad (3)$$

Para estimar el *jitter* no es necesario calcular el valor exacto. Los extremos receptores obtienen un estimado utilizando una formula simplificada [12]. El valor inicial del *jitter* se considera igual a cero.

$$J(i) = J(i - 1) + (|T3(i - 1, i)| - J(i - 1))/16 \quad (4)$$

donde:

$$T3(i - 1, i) = T3_{i-1} - T3_i \quad (5)$$

En el caso de la pérdida de paquetes se realiza el análisis sobre el escenario de la figura 4. Al enviar paquetes de h1 a h2, vía s1 y s2, estos pueden perderse en el camino. Primeramente, se asume que el controlador conoce la topología de la red. De esta forma el controlador envía una consulta a s1 y a s2 respectivamente donde solicita la cantidad de paquetes de interés. Cuando el controlador recibe la respuesta de s1 la almacena en una variable llamada *inpmts*. Al recibir la respuesta de s2 la almacena en otra variable denominada *outpmts*.

Para calcular la pérdida de paquetes PI (%) basta con utilizar la siguiente ecuación [13]:

$$P1 (\%) = \frac{inpmts - outpmts}{inpmts} * 100 \quad (6)$$

Para evaluar la QoS se realiza una comparación entre los valores de los parámetros de QoS obtenidos mediante el análisis del servicio y los valores de los mismos recogidos en documentos o investigaciones que proveen estándares para la calidad de los servicios. Ejemplo de estas investigaciones son la Recomendación G.1010 [14] del Comité de estandarización de la UIT y el documento publicado por Cisco denominado *QoS for VoIP* [15], que constituye una referencia para evaluar la QoS en el servicio VoIP. El resultado de dicha comparación permite establecer el grado de calidad del servicio que se logra en el servicio analizado y, sobre todo, tomar decisiones en cuanto al funcionamiento de la red.

Cuando los parámetros de QoS no están dentro de los límites establecidos se deben aplicar soluciones para corregirlos. Un ejemplo de estas soluciones es la aplicación de mecanismos de QoS [16] como los mencionados anteriormente.

### Estimación y evaluación de QoE

Existen tres clasificaciones de métodos de estimación de QoE: subjetivos, objetivos y correlación entre los objetivos y los subjetivos. Atendiendo a esto se procedió a realizar una comparación entre los métodos de estimación (tabla 1). A partir de esta comparación solo queda decidir, atendiendo al criterio individual el método a escoger.

**Tabla 1. Comparación de los métodos de evaluación de QoE**

Parámetros de comparación	Métodos Objetivos	Métodos Subjetivos	Correlación Objetivo-Subjetivo
Costo de procesamiento de información	Medio	Alto	Medio
Duración en el tiempo	Baja	Alta	Baja
Posibilidad de ser aplicado en tiempo real	Alta	Baja	Alta
Intrusivo	Medio	Bajo	Medio
Aplicable a servicios IP	Sí	Sí	Sí

Luego de llevar a cabo el proceso de aplicación de un método de evaluación de QoE, se procede a valorar este resultado. Por lo general los valores de QoE se presentan en la escala del sistema de medición MOS, el método más difundido para establecer valores de QoE.

### Interpretación de resultados y medidas a aplicar

Si persiste un deterioro en la QoE será necesario primeramente analizar el desempeño del controlador (figura 6) y los *switches*, debido a que existe la posibilidad de que no sean capaces de manejar todas las solicitudes en tiempo por falta de recursos de procesamiento. En caso de que aún continúe degradada la QoE, deben ser analizados otros factores que influyan sobre la misma, como por ejemplo: los humanos y los de contexto.

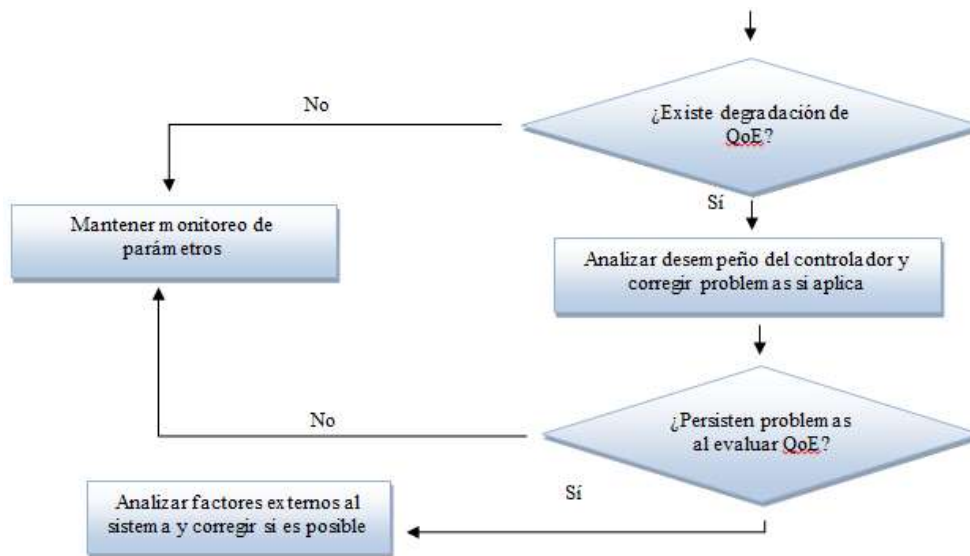


Fig. 6. Análisis de resultados y medidas a aplicar

## RESULTADOS

Empleando el procedimiento propuesto se desarrolló una aplicación externa al controlador empleando el lenguaje de alto nivel de abstracción *frenetic* para la programación de aplicaciones SDN.

La aplicación cuenta con cinco módulos y un programa principal que se encargará de inicializar la aplicación. Cada módulo y su función son listados a continuación:

- Módulo Pérdida: Será el encargado de calcular la pérdida de paquetes.
- Módulo Demora: Calcula la demora en el camino principal por la cual transiten los paquetes.
- Módulo *Jitter*: Será el encargado de calcular el *jitter* basado en los datos del módulo Demora.
- Módulo QoS: Evalúa la QoS basada en la pérdida de paquetes, la latencia y el *jitter*.
- Módulo QoE: Se encarga de evaluar la QoE según el modelo seleccionado.

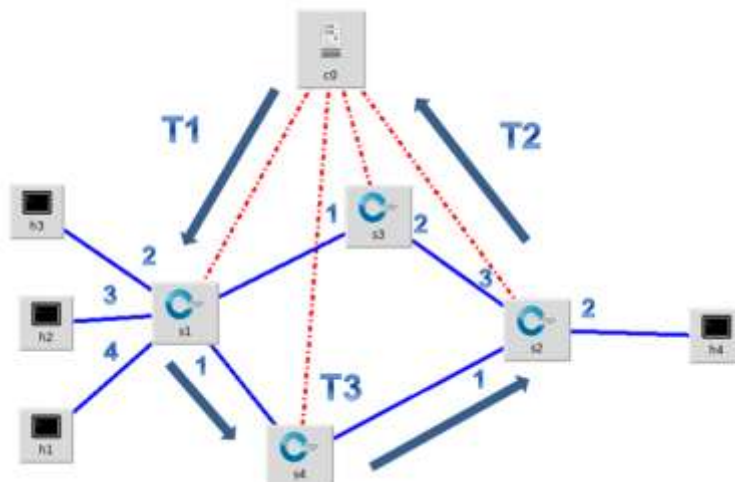
### Fase “Caracterización de la Red y los Servicios”

Sobre la SDN de la figura 7, se simula un servicio de VoIP en un solo sentido empleando VLC, para transmitir un fichero de audio codificado RTP desde h1 hasta h4. El host h4 emplea el puerto 5 000 UDP para la recepción de los paquetes. Las reglas instaladas por defecto en los *switches* permiten solamente el tránsito de los paquetes desde s1 hasta s2 a través de s4, dejando como alternativo el camino a través de s3.

### Fase “Gestión de la QoS”

Los parámetros de QoS seleccionados para ser evaluados son: la latencia, la pérdida de paquetes y el *jitter*.

Para el cálculo de la latencia primeramente se envía una solicitud de tipo *port\_stats* desde el controlador *c0* a *s1* y a *s2* (figura 7). De esta solicitud solo interesa calcular los tiempos *T1* y *T2*. Es por ello que el contenido de la solicitud es desechado. *T2* se obtiene de la misma manera que *T1*.



**Fig. 7. Escenario empleado para el desarrollo de la aplicación**

Luego se envía un paquete de prueba desde *s1-eth1* mediante una solicitud *packet\_out* (figura 7). Cuando el paquete llega a *s2* genera un evento de tipo *packet\_in* y es capturado por el controlador *c0* obteniendo *TT*.

*T3* resulta de la diferencia de sustraer *T1* y *T2* a *TT*. Este proceso se repite cada segundo y los valores de *T3* son almacenados en un arreglo que será utilizado para el cálculo del jitter según la ecuación (4).

La pérdida de paquetes es calculada enviando una *query* del controlador *c0* a *s1-eth4* y a *s2-eth1* que no afecta el flujo de datos y que devuelve los paquetes que pasan por esos puertos y cumplen que su encapsulamiento sea IP, UDP y Puerto Destino igual a 5 000.

En la versión de *frenetic* utilizada no existe aún soporte para QoS. Es por esto que se establece un mecanismo primario para garantizar la misma empleando cambios de camino basados en los valores obtenidos.

Contando con la demora, el *jitter* y la pérdida de paquetes se llama a la función *obtenerQoS* que evalúa la QoS de acuerdo a un conjunto de reglas basadas en estándares y trabajos revisados. La mayor ventaja que tienen estas reglas es que pueden ser modificadas de acuerdo a los requerimientos del servicio que se evalúe.

#### **Fase “Estimación y evaluación de la QoE”**

Para la evaluación de la QoE, después de analizar la comparación realizada en la tabla 1, se escogió un método objetivo, específicamente el referido en la UIT-T G.107 y conocido como Modelo-E, ya que posee las siguientes características que lo hacen idóneo para el servicio VoIP [17]:

- Puede emplearse en VoIP.
- Describe un alto conocimiento de la percepción humana basado en factores de deterioro de la señal de voz.
- Puede modelarse mediante una herramienta de software.
- Provee resultados cuantitativos y fiables de QoE.

#### **Pruebas realizadas**

Con el objetivo de comprobar el funcionamiento de la aplicación realizada se llevaron a cabo pruebas cuyos resultados se discuten a continuación.

Primeramente, se realizó una transmisión sin alterar los parámetros de los enlaces en el trayecto de *s1* a *s2* a través de *s4*. Como se aprecia en la figura 8, de acuerdo con los valores de pérdida, demora y *jitter*, la QoS es evaluada de Buena y el MOS toma valores cercanos a 4, por lo que se puede afirmar, que los usuarios se encuentran satisfechos o muy satisfechos teniendo en cuenta que en condiciones ideales el MOS nunca será superior a 4,5.

```

frenetic@ubuntu-1404: ~/tesis
File Edit Tabs Help
frenetic@ub... x frenetic@ub... x frenetic@ub... x frenetic@ub... x
2017-05-03 14:06:31,150 [INFO] ----- Camino Principal -----
2017-05-03 14:06:31,154 [INFO] Perdida: 0 %
2017-05-03 14:06:31,157 [INFO] Demora: 46.6784238815 ms
2017-05-03 14:06:31,163 [INFO] Jitter: 0.362103058659
2017-05-03 14:06:31,163 [INFO] QoS: Buena
2017-05-03 14:06:31,164 [INFO] QoE: [R: 77.808787599 | MOS 3.93855702703]
2017-05-03 14:06:32,146 [INFO] #####
2017-05-03 14:06:32,150 [INFO] ----- Camino Principal -----
2017-05-03 14:06:32,154 [INFO] Perdida: 0 %
2017-05-03 14:06:32,158 [INFO] Demora: 48.5373735428 ms
2017-05-03 14:06:32,196 [INFO] Jitter: 0.455655971321
2017-05-03 14:06:32,198 [INFO] QoS: Buena
2017-05-03 14:06:32,198 [INFO] QoE: [R: 77.7506694761 | MOS 3.93622165625]
2017-05-03 14:06:33,145 [INFO] #####
2017-05-03 14:06:33,147 [INFO] ----- Camino Principal -----
2017-05-03 14:06:33,149 [INFO] Perdida: 0 %
2017-05-03 14:06:33,154 [INFO] Demora: 26.5284906464 ms
2017-05-03 14:06:33,180 [INFO] Jitter: 0.948877707913
2017-05-03 14:06:33,183 [INFO] QoS: Buena
2017-05-03 14:06:33,184 [INFO] QoE: [R: 78.4578468622 | MOS 3.96436889814]
2017-05-03 14:06:34,145 [INFO] #####
2017-05-03 14:06:34,146 [INFO] ----- Camino Principal -----
2017-05-03 14:06:34,149 [INFO] Perdida: 0 %
2017-05-03 14:06:34,198 [INFO] Demora: 54.0264844894 ms
2017-05-03 14:06:34,201 [INFO] Jitter: 0.829551764024
2017-05-03 14:06:34,202 [INFO] QoS: Buena
2017-05-03 14:06:34,203 [INFO] QoE: [R: 77.5800166222 | MOS 3.92934429474]
    
```

Fig. 8. Evaluación de QoS/QoE en el camino de s1-s2

Como se puede apreciar todos los parámetros están dentro de los rangos permisibles y por tanto los resultados son positivos.

Luego se establece la pérdida de paquetes a 3 % en el enlace s1-s4 a un 3 % (figura 9).

```

frenetic@ubuntu-1404: ~/tesis
File Edit Tabs Help
frenetic@ub... x frenetic@ub... x frenetic@ub... x frenetic@ub... x
2017-05-03 14:21:08,632 [INFO] QoE: [R: 0.523031549998 | MOS 0.996644167022]
2017-05-03 14:21:08,719 [INFO] Prueba: 16 %
2017-05-03 14:21:09,572 [INFO] #####
2017-05-03 14:21:09,572 [INFO] ----- Camino Principal -----
2017-05-03 14:21:09,574 [INFO] Perdida: 3 %
2017-05-03 14:21:09,578 [INFO] Demora: 0.356435775757 ms
2017-05-03 14:21:09,579 [INFO] Jitter: 1.16983151885
2017-05-03 14:21:09,580 [INFO] QoS: Regular
2017-05-03 14:21:09,580 [INFO] QoE: [R: 40.5100104988 | MOS 2.08906241682]
2017-05-03 14:21:09,581 [INFO] ----- Camino Secundario -----
2017-05-03 14:21:09,606 [INFO] Perdida: 16 %
2017-05-03 14:21:09,609 [INFO] Demora: 27.1805524826 ms
2017-05-03 14:21:09,609 [INFO] Jitter: 1.56538122246
2017-05-03 14:21:09,609 [INFO] QoS: Mala
2017-05-03 14:21:09,612 [INFO] QoE: [R: 3.5581739537 | MOS 0.988957316060]
2017-05-03 14:21:09,669 [INFO] Prueba: 20 %
2017-05-03 14:21:10,570 [INFO] #####
2017-05-03 14:21:10,571 [INFO] ----- Camino Principal -----
2017-05-03 14:21:10,573 [INFO] Perdida: 4 %
2017-05-03 14:21:10,577 [INFO] Demora: 40.6229496002 ms
2017-05-03 14:21:10,579 [INFO] Jitter: 1.4286900651
2017-05-03 14:21:10,579 [INFO] QoS: Mala
2017-05-03 14:21:10,580 [INFO] QoE: [R: 32.2164045249 | MOS 1.70286839108]
2017-05-03 14:21:10,581 [INFO] Cambiando a Camino Secundario !!
2017-05-03 14:21:10,611 [INFO] ----- Camino Secundario -----
2017-05-03 14:21:10,613 [INFO] Perdida: 0 %
2017-05-03 14:21:10,616 [INFO] Demora: 25.0661373138 ms
2017-05-03 14:21:10,616 [INFO] Jitter: 0
    
```

Fig. 9. Evaluación de QoS/QoE con 3 % de pérdida de paquetes entre s1-s4



En esta figura se puede apreciar que si la QoS es Regular (pérdida de paquetes =3 % entre s1-s4) se comienzan a enviar paquetes de prueba por el camino alternativo (s1-s2 a través de s3), cuyo objetivo es evaluar si este camino cumple con los requerimientos de QoS. La elevada pérdida de paquetes en el camino alternativo que se aprecia es debido a que esta va disminuyendo a medida que se envían los paquetes de prueba del camino alternativo. Sin embargo, si en un instante la pérdida de paquetes alcanza un valor de 4 %, automáticamente se cambia al camino secundario y se dejan de enviar los paquetes de prueba.

#### **Fase “Interpretación de los resultados y medidas a aplicar”**

Con las pruebas realizadas hasta el momento, la QoE puede ser mejorada gestionando la QoS mediante el cambio de trayectos, por lo tanto, resulta innecesario evaluar el desempeño del controlador y analizar otros factores externos al sistema. Sin embargo, en caso de que fuera necesario se recomienda la utilización de la aplicación *cbench* debido a su simplicidad y facilidad para realizar las pruebas.

Si la QoS es Buena se mantiene el flujo normal de paquetes de s1 a s2 a través de s4. Si es Mala automáticamente se cambia el camino a través de s3 o camino secundario. Una vez que la QoS se vió afectada no va a ser determinante si el camino a través de s3 es elegible o no porque se está evaluando *un servicio en tiempo real* sobre el cual se desconoce su duración. Sin embargo, si la QoS es Regular entonces se obtienen nuevamente los parámetros de QoS del camino alternativo y se comparan con los del camino principal. En caso de que los valores de los parámetros del camino alternativo cumplan con los requerimientos del servicio se procede a cambiar de camino.

#### **CONCLUSIONES**

La estimación y gestión de la QoE cobra cada vez más importancia para los proveedores de servicios de telecomunicaciones ya que esta constituye un reflejo de la aceptación u opinión que tienen los usuarios sobre los mismos.

En este artículo se evalúa la opinión subjetiva de los usuarios basada en parámetros de QoS para una SDN aprovechando el principio de funcionamiento de las mismas. Esto permite la toma de decisiones con el objetivo de prevenir el rechazo del servicio por parte de dichos usuarios. Para ello, algunos de los aspectos más importantes a tener en cuenta son: la optimización y expansión de la infraestructura de red y las características y funcionalidades del equipamiento empleado. Además, se comprobó que es posible reaccionar ante un deterioro de la QoE mediante el enrutamiento de flujos. Por otra parte, se evidenciaron las limitaciones que presenta *frenetic* en cuanto a capacidades de QoS que pueden ser suplementadas mediante el desarrollo de funcionalidades para este fin.

#### **REFERENCIAS**

1. QUALINET. Qualinet White Paper on Definitions of Quality of Experience. Output version of the Dagstuhl seminar 12181. Disponible en: Web: <http://www.qualinet.com>. June, 2012.
2. ITU-T. How to increase QoS/QoE of IP-based platform(s) to regionally agreed standards, March, 2013.
3. Fernández J. Algoritmo de Calidad de Experiencia para transmisiones de video en Redes Definidas por Software, 2014.
4. Németh F, Sonkoly B, Gulyás A. Towards SmartFlow: Case Studies on Enhanced Programmable Forwarding in OpenFlow Switches, 2012.
5. Gubbi J, Marusic S, Palaniswami M. Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions, 2013.
6. Mas M. Arquitectura para la mejora de la calidad de servicios multimedia en Redes Definidas por Software, 2015.
7. Egilmez H, Civanclar S, Tekalp A. An Optimization framework for qos-enabled adaptive video streaming over openflow networks. Multimedia, IEEE Transactions on, 2013. 15 (3):710-715. DOI:10.1109/TMM.2012.2231645. 2013.
8. Duan Q, Wang C, Li X. End-to-end Service delivery with qos guarantee in software defined networks. ARXIV:1504.04076, 2015.
9. Mirchev A. Survey of concepts for qos improvements via sdn. seminar future internet, 2015.
10. D'souza D, et al. Improving qos in a software-defined network. Capstone Research Paper. April 30, 2016.
11. Phemius K. Monitoring latency with OpenFlow. 9th CNSM and Workshops, 2013.

12. Dainese A. Network ping with delay, jitter and MOS, 2014.
13. Ke Ch. Traffic measurement. 2013.
14. UIT. Recomendación UIT-T G.1010. Serie G: sistemas y medios de transmisión, sistemas y redes digitales. Calidad de servicio y de transmisión. Categorías de calidad de servicio para los usuarios de extremo de servicios multimedios, 2001.
15. CISCO. QoS for VoIP, 2001.
16. Elisangela B, Aguiar S, et al. Trends and Challenges for Quality of Service and Quality of Experience for Wireless Mesh Networks, 2011.
17. Assem H, Dunne J, et al. Monitoring VoIP Call Quality Using Improved Simplified E-model, 2013