

PROPUESTA PARA LA EVALUACIÓN DE PARÁMETROS DE QOS EN SDN

Ing. Cesar Alonso Irizar¹, Dr. C. Caridad Anías Calderon²

¹Profesor instructor CUJAE, dirección, ²Profesora titular CUJAE, dirección
¹e-mail: cesar.860709@gmail.com

RESUMEN

En este trabajo, basado en el principio de funcionamiento de SDN y en el soporte de OpenFlow, se propone un procedimiento que indica los pasos a seguir para evaluar parámetros de Calidad de Servicio (Quality of Service, QoS) en las Redes Definidas por Software (Software Defined Networks, SDN) con el propósito de tomar acciones en caso de que la evaluación resulte en deterioro del servicio que se analiza. Para la validación de dicho procedimiento se desarrolla de una aplicación SDN que permite obtener y evaluar parámetros QoS, empleando para ello el lenguaje de alto nivel de abstracción Frenetic. La aplicación desarrollada permite obtener, como parámetros de QoS, la latencia, el *jitter* y la pérdida de paquetes y estimar los tiempos de respuesta del controlador mediante funciones específicas.

PALABRAS CLAVE: QoS, SDN, evaluación, parámetros de desempeño, procedimiento

ABSTRACT

Based on the SDN operation principle and the OpenFlow support, in this work a procedure is proposed with the steps to follow to evaluate QoS parameters on SDN in order to take actions if service is degraded. To validate this procedure an SDN application developed for obtaining and evaluate QoS parameters using Frenetic as a high level abstraction language The developed application obtains delay, jitter, packet lost and estimates the controller reply time through specific functions.

KEYWORDS: QoS, SDN, evaluation, performance parameters, procedure.

INTRODUCCIÓN

En la actualidad las empresas proveedoras de servicios y los fabricantes de equipos de telecomunicaciones están enfrentando una multitud de retos. El desarrollo y la evolución de las redes a otros paradigmas como SDN, conlleva la creación de nuevos métodos de gestión que lo aprovechen. Sin embargo, se mantiene el objetivo de brindar un servicio de calidad para los usuarios finales. Para lograr esto, y minimizar el impacto negativo que puede ocasionar la diversidad y cantidad de tráfico en la red en el servicio que se ofrece, es muy importante la gestión de los dispositivos y los enlaces que intervienen en dicho servicio. Dado que los parámetros de QoS se han estandarizado y se encuentra definido el rango de valores permitidos para cada servicio, resulta suficiente identificar el servicio a evaluar y, basado en la tecnología de red en cuestión, obtener, a través de alguna aplicación, los valores de los parámetros de QoS para, en el caso de que estos no fueran aceptables, tomar decisiones con vistas a mejorarla.

Adicionalmente, el surgimiento de las SDN ha permitido que el desarrollo de aplicaciones para las mismas resulte una opción válida, más aún si se emplea algún lenguaje de alto nivel de abstracción, aunque, en ocasiones, la documentación existente sobre estos no es suficiente.

En este trabajo se propone un procedimiento que indica los pasos a seguir para obtener y evaluar parámetros QoS en redes SDN, empleando mensajes OpenFlow entre el controlador SDN y los *switches* que componen la red. Este procedimiento se valida con el desarrollo de una aplicación SDN compleja y externa al controlador SDN, empleando para ello un lenguaje de alto nivel de abstracción.

MATERIALES Y MÉTODOS

Para sustentar la calidad de servicio (QoS) extremo-extremo es importante profundizar en el comportamiento dinámico de las redes, lo que se hace a través de parámetros que pueden ser medidos y monitoreados. Entre los parámetros más importantes, que determinan si se cumple con el nivel de servicio que se ofrece, se encuentran el retardo, el *jitter* y la pérdida de paquetes [1, 2].

La literatura científica a destacar, relacionada con la evaluación de la QoS y la toma de decisiones en caso de que esta se vea afectada, es:

El trabajo de [3] propone un esquema de calidad de servicio SDN que considera los parámetros tradicionales para medir el rendimiento (caudal eficaz, tasa de transmisión, retardo, *jitter*), pero solo para la red de acceso de banda ancha, como por ejemplo ADSL (*Asymmetric Digital Subscriber Line* por sus siglas en inglés).

En [4] se diseña un controlador que permite obtener calidad de servicio (QoS) para la entrega de servicios multimedia a través de redes que emplean el protocolo OpenFlow. Para ello los flujos multimedia se colocan dinámicamente en rutas que garantizan la calidad de servicio, y que provocan efectos adversos mínimos sobre otros tipos de tráfico en la red. El resultado es comparado con modelos de QoS para las redes convencionales como IntServ y DiffServ en cuanto a: soporte de flujos, tipos de garantía, complejidad, efecto en otros flujos y mecanismos para brindar QoS.

Por su parte [5] propone una modificación al controlador Floodlight que brinda soporte para QoS permitiendo establecer valores en el campo Tipo de Servicio (*Type of Service*, ToS por sus siglas en inglés) del datagrama IP con el fin de brindar tratamiento diferenciado mediante colas a los paquetes.

Los autores de [6] sugieren la creación de un *framework* que se basa en el principio de Red como Servicio (*Network as a Service*, NaaS por sus siglas en inglés) para brindar una abstracción de red y una orquestación para aprovisionamiento de servicios extremo-extremo. En el mismo se propone una técnica para determinar los requerimientos de ancho de banda que garanticen QoS.

La propuesta de [7] se apoya en las opciones que brinda el protocolo OpenFlow en su versión 1.0 para brindar QoS mediante encolado y marcado de paquetes empleando el campo ToS de IP. Además utiliza un módulo que emplea el algoritmo de Dijkstra para garantizar las QoS del tráfico. El controlador empleado es Floodlight.

Basado en Floodlight en [8] se desarrolla un módulo externo que permite la configuración de colas en OpenvSwitch, algo que comúnmente se encuentra fuera del alcance de los controladores. Este módulo se soporta en las API REST y puede extenderse a otros controladores.

Teniendo en cuenta la revisión realizada anteriormente, queda claro que algunos puntos importantes que deben tomarse en cuenta al trabajar con QoS en SDN son:

- De forma general, los controladores SDN tienen capacidades de QoS limitadas o nulas. Por esto, las propuestas existentes implementan un módulo adicional en el controlador para proporcionar la QoS. Sin embargo, suelen ser variadas las propuestas, lo que trae consigo que, junto al emergente desarrollo de esta tecnología, no exista un criterio unificado sobre cómo proceder para obtener y evaluar parámetros de QoS. El protocolo OpenFlow también tiene un soporte limitado de calidad de servicio. Típicamente se emplean colas simples para reservación de la tasa de transmisión, y es posible monitorear la pérdida de paquetes.
- El enfoque de gran parte de los trabajos publicados va dirigido al enrutamiento y a la priorización de los flujos para mejorar la QoS.

PROPUESTA

Con el objetivo de evaluar la QoS en SDN para servicios IP, teniendo en cuenta lo antes planteado, se propone el procedimiento que se muestra en la figura 1. La caracterización de la red y los servicios es el primer bloque de la propuesta y posee gran importancia, pues de su correcta realización dependerá que el resto del procedimiento fluya adecuadamente y que los resultados obtenidos se correspondan con los esperados de acuerdo al escenario que se analice.

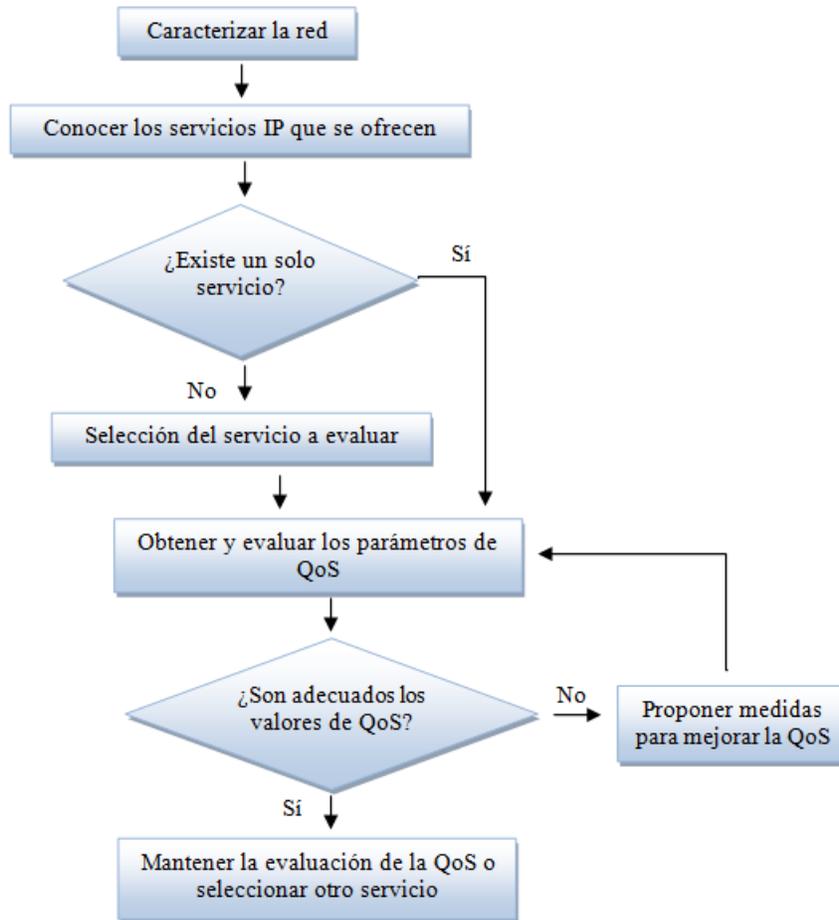


Figura 1: Procedimiento para la gestión de la QoS en servicios IP.

Caracterizar la red no es más que ofrecer una descripción global de la composición, disposición y función de los elementos que la conforman [9] para así tener conocimiento del tipo de red en la que se actúa y, con ello, a priori, comprender qué tipos de servicios se pueden brindar sobre ella.

El conocimiento de los servicios IP es muy importante pues indica los tipos de tráficos presentes en la red que se analiza y permite clasificarlos. En este análisis se debe hacer un importante énfasis en los servicios multimedia, ya que son los que mayores tráficos agregan a la red, encontrándose, además, entre los que más requerimientos tienen.

La elección de un servicio está dada por el interés particular que se tenga en evaluar el desempeño del mismo. Se recomienda, como parte de la propuesta, centrarse en la evaluación de uno solo cada vez, debido a que la obtención de valores de los parámetros deseados podría complejizarse, con las herramientas existentes en la actualidad, al evaluar varios servicios a la vez.

En este trabajo se propone seleccionar como parámetros de evaluación de la QoS la demora o latencia, el *jitter* y la pérdida de paquetes.

Para obtener los valores de la demora o latencia se aprovecha el principio de operación de las SDN. En otras palabras, la información sobre un camino se captura directamente de los dispositivos de red. La idea principal es usar los mensajes de OpenFlow para medir la demora entre los *switches*.

Al calcular la latencia, el primer paso consiste en crear un paquete que será usado como prueba [10]. El controlador pide al *switch* s1 enviar el paquete a través de un puerto al próximo *switch* s2 y se registra el tiempo de envío. Cuando el *switch* s2 recibe el paquete, envía un mensaje al controlador para comunicarle el estado del paquete y se registra el tiempo de recepción. La figura 2 ilustra el mecanismo descrito.

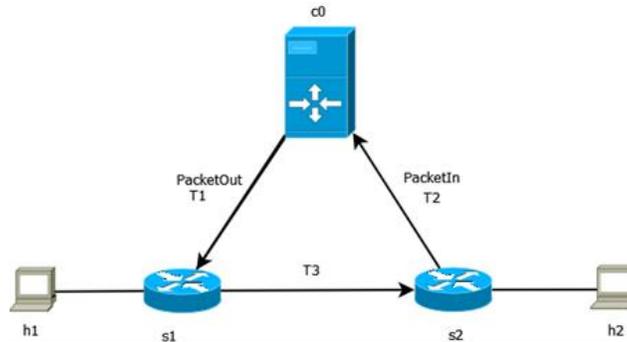


Figura 2: Escenario para evaluar parámetros de QoS

La demora necesaria corresponde a T3, que es el tiempo que demora el paquete en llegar de un *switch* a otro. Cuando el controlador recibe el mensaje, el tiempo total puede ser hallado con la expresión 1.

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

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.

Para calcular tanto T1 como T2 se tiene en cuenta el RTT (*Round Trip Time* por sus siglas en inglés) que resulta de enviar una pregunta y recibir una respuesta del controlador al *switch* y viceversa y se emplea la expresión 2.

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

Ta es el instante en que se envía el paquete de prueba que en este caso es una solicitud OpenFlow de tipo *ports-stats-request* y Tb es el instante en que se recibe la respuesta al mensaje que es de tipo *port-stats-received* [11]. El mismo método puede aplicarse a T2. En consecuencia, se obtiene T3 despejándolo de la expresión 1 (expresión 3).

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

Para estimar el *jitter* se obtiene en los extremos receptores un estimado del mismo utilizando la ecuación simplificada 4 [12].

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

donde:

i: valor anterior del parámetro calculado en la evaluación anterior

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

La misma calcula el *jitter* de forma iterativa. El valor inicial del *jitter* se considera igual a cero.

Para la evaluación de la pérdida de paquetes se emplea nuevamente el escenario de la figura 2.

Al enviar paquetes de h1 a h2, vía s1 y s2, estos pudieran perderse en el camino. Se asume que el controlador conoce la topología de la red lo que permite que el controlador envíe una encuesta a s1 y a s2 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*.

La pérdida de paquetes *Plost* (%) se puede obtener empleando la ecuación 6 [13]:

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

donde:

inpmts: cantidad de paquetes que entran al switch que cumplen con las condiciones que se encuestan.

outpmts: cantidad de paquetes que salen del switch que cumplen con las condiciones que se encuestan.

La QoS se evalúa realizando una comparación entre los parámetros obtenidos mediante el análisis del servicio y los valores de los mismos recogidos en documentos que proveen estándares para la calidad de los servicios. Ejemplo de estos documentos son la Recomendación G.1010 [2] del Comité de estandarización de la UIT (UIT-T, por sus siglas en inglés) que constituye una referencia para evaluar la QoS en servicios multimedia, aunque la misma aclara que en ocasiones puede resultar difícil alcanzar los valores recomendados. Es por ello que, como parte de los SLA (*Service Level Agreement* por sus siglas en inglés) que se pactan con el cliente, se definan los OLA (*Operational Level Agreement* por sus siglas en inglés), a partir de los cuales se negocian de forma medible los valores de los parámetros de QoS a evaluar. El

resultado de esta comparación permite establecer el grado de calidad de servicio que se logra en el servicio analizado y, sobre todo, tomar decisiones, si es necesario, para mejorar el desempeño de la red y sus servicios. En la tabla 1 se muestran los requerimientos de los parámetros de QoS para los servicios de video.

Tabla 1: Parámetros de QoS para servicios de video

Parámetros de QoS	Latencia (ms)	Jitter (ms)	Pérdida de paquetes (PLR) (%)
Buena	< 150	< 1	< 2
Media	150-400	1 - 3	2 - 4
Pobre	> 400	> 3	> 4

RESULTADOS

Para validar el procedimiento propuesto, se desarrolla, empleando el lenguaje de alto nivel Frenetic, una aplicación SDN que permite obtener y evaluar los parámetros de QoS latencia, *jitter* y pérdida de paquetes y estimar los tiempos de respuesta del controlador mediante funciones específicas.

La selección de Frenetic se basa en la utilidad y facilidad que tiene el mismo, como lenguaje de alto nivel de abstracción, para desarrollar aplicaciones para SDN. La base de Frenetic es Python, lo que permite a los programadores que estén familiarizados con el funcionamiento de este lenguaje de programación, emplear las funciones, ya implementadas, que pudieran ser reutilizadas o modificadas a conveniencia [14].

Para crear el escenario de red a validar (figura 3) se empleó Miniedit, una interfaz gráfica de Mininet. Dicho escenario está compuesto por cuatro *switches*, cuatro *hosts* y un controlador SDN. Los *switches* utilizan Open vSwitch, mientras que el controlador empleado es Frenetic.

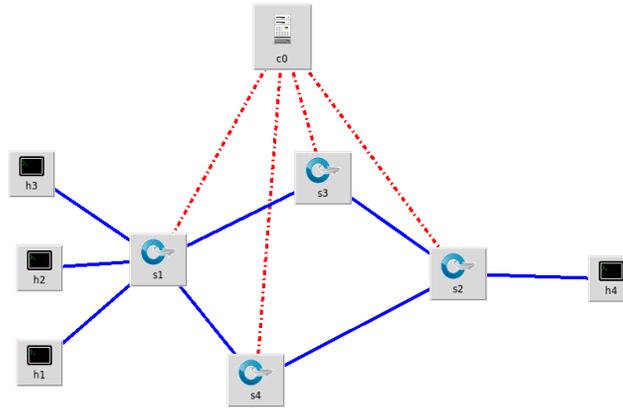


Figura 3: Escenario para obtención de parámetros de QoS

Los switches s1, s4 y s2 tienen políticas previamente instaladas que permiten la comunicación h1-h4, h2-h4, h3-h4 y viceversa, solo a través de s4, dejando como alternativo el camino a través de S3.

En el programa principal de la aplicación desarrollada para validar el procedimiento que se propone, se llaman a funciones que actualizan las tablas de flujos de los *switches* y calculan y evalúan los parámetros de Qos.

Se ha considerado que el servicio a evaluar en el escenario propuesto es el de *streaming*. Para la obtención de los parámetros Qos se debe tener en cuenta los datos propios del servicio, tales como IP fuente, puerto fuente, IP destino, puerto destino, dado que los paquetes deben ser previamente identificados (figura 4) para proceder a obtener sus parámetros. Además, se ha considerado en la simulación que los paquetes UDP van destinados al puerto 5000.

```
def politica_switch1(self):
    return IfThenElse(
        SwitchEq(1) & PortEq(4) & IPProtoEq(17) & TCPEq(5000),
        SendToQuery("sw1" | SetPort(1),
        self.politica_repetidor()
    )

def politica_switch2(self):
    return IfThenElse(
        SwitchEq(2) & PortEq(1) & IPProtoEq(17) & EthTypeEq(0x800) & TCPEq(5000),
        SendToQuery("sw2" | SetPort(2),
        self.politica_repetidor()
    )
```

Figura 4: Función para identificar y enviar los paquetes deseados

La latencia se obtiene como resultado de evaluar una función donde t1 y t2 son los tiempos de un solo sentido cuando el controlador solicita información al switch 1 y al switch 2 respectivamente. El *jitter* se

obtiene mediante una función que emplea la expresión 4 analizada al explicar el procedimiento que se propone y los valores obtenidos de latencia.

Para obtener la pérdida de paquetes se encuesta la cantidad de paquetes que llegan al *switch* 1 puerto 4 con puerto destino UDP igual a 5000. A continuación se encuesta la cantidad de paquetes que llegan al *switch* 2 puerto 1 con puerto destino udp igual a 5000. Luego se calcula el porcentaje que representan los paquetes que llegan con estas características al *switch* 2 sobre el total de paquetes que envía el *switch* 1.

Finalmente, la Qos se evalúa empleando las funciones que se recogen de la figura 5. Resulta válido destacar que se han incluido un conjunto de reglas generales en la lógica de decisión para evaluar la Qos que pueden ser modificables de forma simple de acuerdo al servicio en cuestión.

```
def imprimirQos(self):
    resultado = self.qos.obtenerQos(self.nib.getLatencia(), self.nib.get_loss(), self.nib.getJitter())
    logging.info("QoS: "+ resultado)

def obtenerQos(self, latencia, perdida, jitter):
    try:
        if latencia < 150:
            if perdida < 2:
                return 'Buena';
            elif perdida > 2 and perdida < 4:
                return 'Buena';
            else:
                return 'Regular';
        elif latencia > 150 and latencia < 400:
            if perdida < 2:
                if jitter < 1:
                    return 'Regular';
                if jitter > 1 and jitter < 3:
                    return 'Regular';
                else:
                    return 'Mala';
            elif perdida > 2 and perdida < 4:
                if jitter < 1:
                    return 'Regular';
                if jitter > 1 and jitter < 3:
                    return 'Mala';
                else:
                    return 'Mala';
            else:
                return 'Mala';
        else:
            return 'Mala';
    except:
        return 'Parametros insuficientes';
```

Figura 5: Funciones para evaluar la QOS

Si al evaluar la QoS los resultados no son satisfactorios, de acuerdo al procedimiento se deben proponer medidas para mejorar la QoS, sin embargo, en la versión de Frenetic utilizada no existe soporte para QoS en SDN [15]. En otras palabras, aunque Frenetic soporta la versión de OpenFlow 1.0, no soporta acciones de encolamiento basado en diferentes tipos de tráfico. Debido a esto se considera, previo conocimiento de la topología, un posible cambio de trayecto en el flujo que se esté analizando. De existir otro camino para el flujo, se debe evaluar la QoS para el nuevo camino siguiendo las pautas anteriormente mencionadas, ya que la elección de un camino o trayecto alternativo no garantiza una mejoría de QoS. Si el camino cumple con los parámetros establecidos para pérdida de paquetes, latencia y jitter entonces se instalan nuevas reglas en los switches que implementen los nuevos requerimientos.

Empleando en el escenario creado, se utilizó el software VLC para transmitir un *streaming* de video desde h1 hasta h4 sin llevar a cabo ninguna acción sobre los dispositivos en el enlace. Al ejecutar la aplicación desarrollada se obtienen los resultados que se muestran en las figuras 6 y 7 donde se observa que los parámetros de Qos no sufren variaciones perceptibles y por tanto la Qos es buena.



Figura 6: Transmisión de video con variaciones no perceptibles de parámetros de QOS

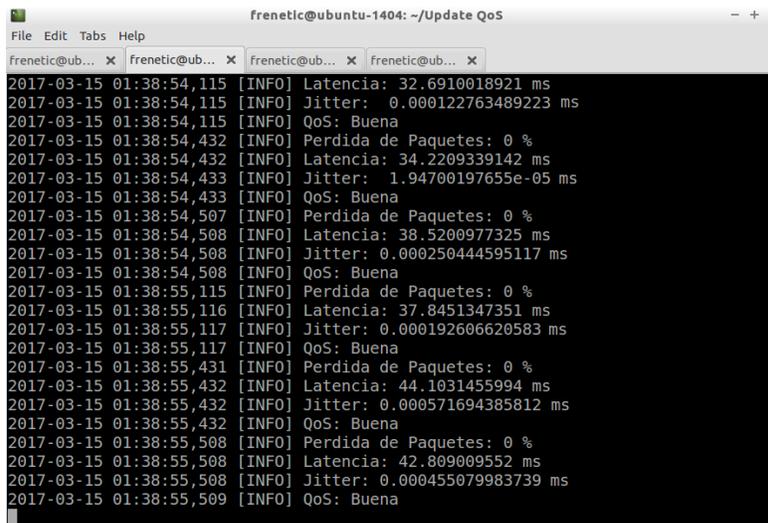


Figura 7: Valores de parámetros de Qos que no afectan la transmisión de video

Sin embargo, introduciendo una latencia media de 250 ms y una pérdida de paquetes de 40% en los dispositivos en Mininet, se puede apreciar que automáticamente a través de la aplicación desarrollada, el controlador comienza a evaluar un camino secundario hasta que hace actualiza las tablas de flujo de los switches y cambia al mismo. Los resultados se muestran en las figuras 8 y 9.

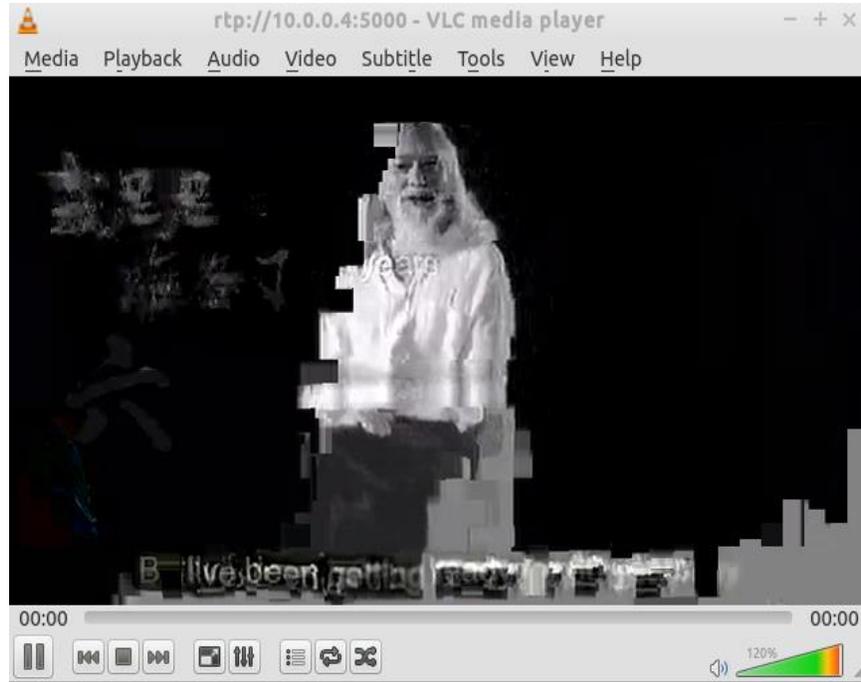


Figura 8: Transmisión de video con variaciones perceptibles en los parámetros de Qos

```
frenetic@ubuntu-1404: ~/salva Marzo 22/tesis
File Edit Tabs Help
frenetic@ub... x frenetic@ub... x frenetic@ub... x
2017-03-26 23:24:28,444 [INFO] -----
2017-03-26 23:24:28,445 [INFO] Perdida de Paquetes: 0 %
2017-03-26 23:24:28,446 [INFO] Demora: 286,305904388 ms
2017-03-26 23:24:28,446 [INFO] Jitter: 0.000276349484921
2017-03-26 23:24:28,446 [INFO] QoS: Regular
2017-03-26 23:24:28,446 [INFO] Evaluando camino alternativo
2017-03-26 23:24:28,450 [INFO] Perdida Camino Secundario: 6 %
2017-03-26 23:24:28,452 [INFO] Demora Camino Secundario: 13 ms
2017-03-26 23:24:28,452 [INFO] Cambiando camino
2017-03-26 23:24:29,444 [INFO] -----
2017-03-26 23:24:29,444 [INFO] Perdida de Paquetes: 0 %
2017-03-26 23:24:29,444 [INFO] Demora: 286,305904388 ms
2017-03-26 23:24:29,444 [INFO] Jitter: 0.000259077642113
2017-03-26 23:24:29,445 [INFO] QoS: Regular
2017-03-26 23:24:29,445 [INFO] Evaluando camino alternativo
2017-03-26 23:24:29,446 [INFO] Perdida Camino Secundario: 3 %
2017-03-26 23:24:29,447 [INFO] Demora Camino Secundario: 13 ms
2017-03-26 23:24:29,447 [INFO] Cambiando camino
2017-03-26 23:24:30,446 [INFO] -----
2017-03-26 23:24:30,446 [INFO] Perdida de Paquetes: 3 %
```

Figura 9: Valores de parámetros de QoS que afectan la transmisión de video y provocan reconfiguración de los switches

CONCLUSIONES

El procedimiento que se propone constituye una forma novedosa para obtener, evaluar y corregir, si aplica, los valores de los parámetros de QoS en redes SDN considera el principio de funcionamiento de estas redes que modifican de cierta manera los paradigmas de gestión clásicos de las redes, ya que mediante una aplicación que se comuniquen con la red a través del controlador SDN esta puede ser gestionada.

El desarrollo de una aplicación SDN, empleando el lenguaje de alto nivel de abstracción Frenetic, para validar el procedimiento propuesto permite obtener y evaluar parámetros QoS. En caso de que resulte deteriorada la QoS la aplicación es capaz, previo conocimiento de la topología de red, de reconfigurar las tablas de flujo si existe otro camino disponible para los paquetes que brinde mejores parámetros de QoS. Además, en la misma es posible cambiar los requerimientos de QoS y las tablas de flujo de los *switches* de acuerdo a los intereses que se tengan. Esta aplicación SDN sirve de base para futuras investigaciones en el campo de QoS y en el propio desarrollo de aplicaciones.

Se recomienda emplear la aplicación desarrollada para evaluar la QoS en diferentes escenarios de redes SDN y para diferentes servicios y que, además, basada en el servicio, sea capaz de hacer una inspección más profunda de los paquetes con el objetivo de determinar el impacto que sobre el servicio tendría el deterioro de algún parámetro de QoS a nivel de aplicación.

REFERENCIAS

1. T. Braun, M. Diaz, J. E. Gabeiras, and T. Staub, End-to-end quality of service over heterogeneous networks: Springer Science & Business Media, 2008.
2. 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.
3. M. S. Seddiki, M. Shahbaz, S. Donovan, S. Grover, M. Park, N. Feamster, et al., "FlowQoS: Per-Flow Quality of Service for Broadband Access Networks," Georgia Institute of Technology, 2015.
4. H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks," in Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific, 2012, pp. 1-8.
5. D. D'zouza, "Improving QoS in a Software-Defined Network", 2016.
6. Q. Duan, C. Wang, and X. Li, "End-to-End Service Delivery with QoS Guarantee in Software Defined Networks," arXiv preprint arXiv:1504.04076, 2015.
7. R. Wallner and R. Cannistra, "An SDN approach: quality of service using big switch's floodlight open-source controller," Proceedings of the Asia-Pacific Advanced Network, vol. 35, pp. 14-19, 2013.
8. D. Palma, J. Goncalves, B. Sousa, L. Cordeiro, P. Simoes, S. Sharma, et al., "The queuepusher: Enabling queue management in openflow," in Software Defined Networks (EWSN), 2014 Third European Workshop on, 2014, pp. 125-126.

9. M. Wijnants, S. Agten, P. Quax, and W. Lamotte, "Investigating the relationship between QoS and QoE in a mixed desktop/handheld gaming setting," in Proceedings of the 5th international student workshop on Emerging networking experiments and technologies, 2009, pp. 29-30.
10. K. Phemius and M. Bouet, "Monitoring latency with openflow," in Network and Service Management (CNSM), 2013 9th International Conference on, 2013, pp. 122-125.
11. Software Defined Network. 2013. [Online]. Available: <http://csie.nqu.edu.tw/smallko/sdn/sdn.htm>. [Accessed: 28-Aug-2017].
12. Network ping with delay, jitter and MOS, 2014. [Online]. Available: <http://www.routereflector.com/2014/08/network-ping-with-delay-jitter-and-mos>. [Accessed: 14-Jul-2017].
13. Traffic measurement, 2013 [Online]. Available: <http://csie.nqu.edu.tw/smallko/sdn/sdn.htm>. [Accessed: 19-Aug-2017].
14. Frenetic-lang/frenetic, 2016. [Online]. Available: <https://github.com/frenetic-lang/frenetic>. [Accessed: 29-Jul-2017].
15. *Frenetic Programmers guide*, 2016. [Online]. Available: https://github.com/frenetic-lang/manual/blob/master/programmers_guide/frenetic_programmers_guide.pdf. [Accessed: 30-Aug-2017].