

Analisis y Monitorización por Agrupamiento de Contenido Multimedia en Redes SDN

Jesús Antonio Puente Fernández and Luis Javier García Villalba, *Member, IEEE*

Resumen—Los proveedores actuales de contenido de vídeo requieren una gran demanda de recursos de red para obtener la mejor calidad de experiencia (QoE) en el lado del usuario. Para obtener la mejor QoE del usuario, la red debe poder anticipar problemas de ancho de banda, contenido congelado, ataques de Denegación de Servicio (DoS), entre otros. Estos problemas no pueden resolverse en las redes IP tradicionales debido a la rigidez de su arquitectura. Teniendo en cuenta los beneficios que proporciona el nuevo concepto de arquitectura de red llamado Redes definidas por software (SDN) se presenta una posible solución para resolver los desafíos de las redes existentes en términos de entrega de contenido de vídeo. Estos aspectos son posibles debido a la separación del plano de control (controlador) y el plano de datos (switches). Usando la vista centralizada de la red que proporciona SDN, es posible desarrollar una herramienta de monitorización que obtenga las métricas de la red que se utilizarán posteriormente. En este artículo se estudia la aplicación de un algoritmo de clustering que optimiza la monitorización del tráfico en las redes SDN disminuyendo el número de consultas de estadísticas para mejorar el tráfico de la red, reducir la sobrecarga y ser utilizadas para ataques DoS. Las simulaciones utilizan un algoritmo de clustering que monitoriza una transmisión de vídeo entre dos hosts demostrando la viabilidad de nuestra propuesta de monitorización.

Palabras claves—clustering, consultas estadísticas, controlador, DoS, monitorización, QoE, seguridad, SDN.

I. INTRODUCCIÓN

Hoy en día, la transmisión de vídeo es una de las aplicaciones más populares y demandadas en Internet. Requiere un gran ancho de banda y altos volúmenes de datos, lo que plantea muchos desafíos para los proveedores de servicios de Internet, que compiten por ser el mejor proveedor en entregar el tráfico de la manera más eficiente posible, al tiempo que mantienen una calidad de servicio alta y continua. La medición de la satisfacción de los usuarios finales con los servicios de red se establece como el concepto de Calidad de Experiencia (QoE).

Trabajos como [1] se centran en comprender y mejorar el concepto de QoE de la transmisión de vídeo adaptativa HTTP (HAS) actual. Esta tecnología permite adaptar la velocidad de bits de vídeo de la red en función de las demandas de transmisión. Se han presentado dos enfoques para monitorear el QoE de HAS: el primer enfoque se encuentra en el trabajo [2] y se usó como ejemplo de monitorización basado en la red [3] [4]. Las métricas de calidad de vídeo se extrajeron del tráfico de la red mediante la inspección profunda de paquetes (DPI). El segundo trabajo se enfoca en monitorizar el QoE de

manera discreta dentro de una aplicación de transmisión de vídeo de YouTube de Android (YoMoApp), proporcionando información precisa sobre los factores de QoE en cómo son percibidos por los usuarios finales [5], [6].

Las Redes Definidas por Software (SDN, del inglés Software Defined Networks) se ha convertido en un concepto para el control dinámico de la configuración de redes informáticas. Desacopla el control de red (plano de control) de los recursos de red subyacentes (plano de datos) controlados a través de un controlador centralizado. También proporciona un conjunto de interfaces de programación de aplicaciones (API) para abreviar la implementación de servicios de red como por ejemplo: enrutamiento, QoS, QoE, DoS y balanceadores de carga entre otros. La combinación de la granularidad de las estadísticas de red y la vista global de la red obtenida en el plano de datos ha creado un interés importante en la monitorización de redes usando SDN. Trabajos como [7] [8] exploran la combinación de protocolos de monitorización tradicionales como NetFlow/IPFIX y sFlow con el protocolo OpenFlow usado en redes SDN.

En este artículo se presenta una discusión de dos algoritmos de clustering que se aplican en el framework de monitorización SDN propuesto en [9]. Se utilizan para disminuir el número de peticiones de monitorización en los switches de la topología sin comprometer la precisión de los valores monitorizados, lo que permite a los administradores de red tener un estado real de los enlaces de red. Para analizar los resultados al aplicar cada algoritmo de clustering, se ha construido un escenario de prueba para transmitir un vídeo entre dos hosts (cliente y servidor) utilizando un simulador de red. Finalmente, los resultados confirman que aplicar diferentes algoritmos de clustering en la monitorización reduce el número de peticiones de estadísticas de red sin pérdida de precisión respecto a las estrategias de monitorización tradicionales.

El resto del documento se describe a continuación: El paradigma de SDN y el uso del protocolo OpenFlow se detallan en la sección II. La sección III describe los trabajos relacionados. El estudio y la discusión de los diferentes algoritmos de clustering se explica en la sección IV. Para verificar los beneficios de cada algoritmo de clustering, la sección V contiene las simulaciones y resultados de las pruebas usando dos topologías. Finalmente, la sección VI concluye con una breve discusión y conclusiones.

J. A. Puente Fernández and L. J. García Villalba. Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial, Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España. e-mail: jesusantoniopuente@ucm.es, javiervg@fdi.ucm.es.

II. REDES DEFINIDAS POR SOFTWARE

Las redes definidas por software (SDN) son un nuevo paradigma de red que gestiona toda la red a través de un controlador de software programable centralizado. Desacopla el plano de control y el plano de datos respecto a las redes tradicionales. Usando la vista centralizada, el controlador puede administrar el tráfico de red de manera más eficiente y dinámica. Además, SDN proporciona una configuración de red dinámica adaptable a la condición de la red o los requisitos del usuario. La lógica de control es programable a través del controlador que envía paquetes de control a los switches de red utilizando el protocolo más extendido para redes SDN, que es OpenFlow [10] [11]. La arquitectura OpenFlow [10] sigue el principio SDN desacoplando el plano de datos y de control. Se basa en tres entidades principales: un switch OpenFlow (plano de datos), un controlador externo (plano de control) y el Protocolo OpenFlow [10]. Por lo tanto, el comportamiento de la red es altamente personalizable según el propósito o los requisitos del usuario. SDN se divide en tres capas lógicas:

1. **Capa de datos (recursos de red):** Está conectada a la capa superior, que es la capa de control a través de las interfaces Southbound. Estas interfaces proporcionan la abstracción de los switches programables y la conexión con el software que se ejecuta dentro del controlador (OpenFlow es el ejemplo más representativo de estas interfaces). Como se mencionó anteriormente, en estas interfaces se ejecuta un Sistema Operativo de Red (NOS) que es responsable de controlar el comportamiento de la red. Ejemplos de NOS son: NOX [12], Floodlight [13] y OpenDaylight [14] entre otros.
2. **Capa de control (controlador SDN):** Está conectada a la capa de aplicación a través de las interfaces Northbound que permiten la creación de aplicaciones y políticas de red de alto nivel que se envían al NOS. Del mismo modo, ejemplos de interfaces en dirección norte son: *Frenetic* [15], *Procera* [16], *Netcore* [17] y *McNettle* [18].
3. **Capa de aplicación:** son programas que comunican comportamientos y recursos necesarios con el controlador SDN a través de la API Northbound. Además, las aplicaciones pueden construir una vista abstracta de la red mediante la recopilación de información del controlador para la toma de decisiones.

Dado que el hardware de reenvío está separado y gestionado por el controlador, los administradores de la red pueden diseñar e implementar cualquier lógica de control utilizando el estándar OpenFlow [10], que es el más extendido en las redes SDN. De este modo, SDN proporciona flexibilidad en el desarrollo de nuevas estrategias y lógicas de red, mientras que las redes tradicionales son limitadas. Varias encuestas y trabajos sobre SDN se centran en la perspectiva de red [19]–[21] en comparación con las redes tradicionales.

OpenFlow [10] se considera uno de los primeros estándares SDN. Originalmente definió el protocolo de comunicación en entornos SDN que permite que el controlador SDN interactúe directamente con el plano de reenvío de dispositivos de red como switches y routers, tanto físicos como virtuales (basados en

hipervisores), para que pueda adaptarse mejor a los requisitos comerciales. Dicho protocolo define la comunicación entre los dispositivos de red de Capa 2 (switches) y el controlador. En una vista más profunda, proporciona la capacidad de programar las tablas de flujo ubicadas dentro de un conmutador y también puede cambiar o introducir una nueva funcionalidad en tiempo de ejecución.

III. ESTADO DEL ARTE

La monitorización de la red es el uso de un sistema que supervisa constantemente una red en busca de componentes lentos o defectuosos y lo notifica al administrador de la red en caso de interrupciones u otros problemas. La supervisión de la red es parte de la administración de la red, ya que se utiliza para recuperar una vista global del tráfico que ingresa a la red. Dichas herramientas son muy útiles para que los administradores de red analicen la red para prevenir futuros problemas de congestión, flujos de tráfico ilegítimos, entre otros problemas.

Mientras que un Sistema de Detección de Intrusos (IDS) monitoriza una red en busca de amenazas del exterior, un sistema de monitorización de red monitoriza el estado de la red en busca de problemas causados por servidores sobrecargados o bloqueados, conexiones de red u otros dispositivos que utilizan diferentes métricas entre otros. Las métricas de medición comunes son el tiempo de respuesta y la disponibilidad, aunque las métricas de consistencia y confiabilidad están comenzando a ganar gran popularidad. La incorporación generalizada de dispositivos de optimización WAN está teniendo un efecto adverso en la mayoría de las herramientas de monitorización de red, especialmente cuando se trata de medir retrasos precisos de extremo a extremo porque limitan la visibilidad del tiempo de retraso de ida y vuelta. Los fallos de solicitud de estado, como cuando no se puede establecer una conexión, tiempos de espera agotados o no se puede recuperar el documento o mensaje, generalmente producen una acción desde el sistema de monitorización.

Los métodos activos y pasivos son dos enfoques para monitorizar redes:

- **Métodos activos:** métodos que envían paquetes a dispositivos de red como switches, servidores, entre otros. Por lo tanto, dado que estos métodos inyectan paquetes en la red, impactan directamente en el tráfico de la red.
- **Métodos pasivos:** métodos compatibles con hardware de propósito especial (sniffers) o integrados en otros dispositivos como routers, switches o hosts. De forma contraria a los métodos activos, no aumentan el tráfico dentro de la red cuando miden estadísticas.

Con respecto al uso de métodos pasivos en redes tradicionales, los protocolos como el Protocolo de administración de red (SNMP) [22] y el Protocolo de configuración de red (NETCONF) [23] permiten monitorizar los dispositivos y las estadísticas de la red. De manera similar, las herramientas de monitorización como NetFlow [24] y sFlow [25] se usan para estimar tanto una muestra como estadísticas completas de tráfico en redes basadas en flujo. jFlow [26] es una extensión del lenguaje Java [27] que permite verificar las anotaciones de

flujo de forma estática. Además, proporciona un compilador que verifica (también estáticamente) y evita fugas de información a través de canales de almacenamiento. jFlow proporciona características que hacen que la comprobación de flujo no sea tan restrictiva como otros lenguajes de programación.

Los métodos pasivos y activos para medir el rendimiento de la red se utilizan en [28]. Utiliza beacons para enviar paquetes sonda e instalar flujos adicionales en los switches. Luego, estos beacons se utilizan para estimar la tasa de pérdida de paquetes y el retraso. Una solución híbrida entre métodos pasivos y activos es el framework propuesto en [29]. Este trabajo define un framework de monitorización de red basado en un módulo orquestador con un método flexible para recuperar estadísticas de red.

OpenNetMon [30] monitoriza las métricas por flujo en las redes OpenFlow. En lugar de utilizar una estrategia única para la monitorización, la aplicación de diferentes métodos para consultar estadísticas puede ayudar a reducir la sobrecarga en los switches y en las redes. En este orden, OpenTm [31] propone seguir una distribución de consulta no uniforme con respecto a un esquema uniforme. Demuestra que esta estrategia es mucho más rápida que las formas existentes de estimación del tráfico en las redes IP. Chowdhury et al. [32] propone Payless, un framework de monitorización eficiente de bajo coste de redes SDN. Su característica principal es que proporciona una vista abstracta de la red y una manera regular de solicitar estadísticas sobre los recursos de la red. Además, dado que se ha desarrollado como un conjunto de componentes conectables, proporciona interfaces para conectarlos a todos (API RESTful de alto nivel).

Continuando con la monitorización de redes de flujo, Flowsense [33] es un enfoque que busca estadísticas de alta precisión con un coste cero de medición utilizando la separación física del plano de control y de datos en SDN. Las características clave son: la duración que tarda el flujo en la entrada de la tabla de flujo, la cantidad de tráfico que coincide con ese flujo y, finalmente, el puerto de entrada del tráfico que coincide con la entrada. Sin embargo, tiene algunas limitaciones, ya que depende en gran medida del tipo de tráfico que va a monitorizar, ya que los flujos grandes como una transmisión de vídeo pueden retrasar su cálculo y su utilización.

Suárez-Varela et al [34] propone un sistema de monitorización habilitado para SDN y reconocimiento de aplicaciones. Este sistema de monitorización está dividido en dos bloques. El primer bloque, llamado SBAR, es un sistema basado en OpenFlow que produce informes de monitorización a nivel de flujo. El segundo bloque es una herramienta de análisis de datos que procesa los informes resultantes de SBAR y muestra estadísticas valiosas de red a través de una interfaz gráfica de usuario web. SBAR, a su vez, está compuesto por un módulo de medición que se encarga de mantener las mediciones de flujo en los switches e informarlas a los controladores, y un módulo de clasificación, que produce etiquetas de clasificación para cada flujo en los informes de medición.

IV. ESTUDIO DE ALGORITMOS DE CLUSTERING PARA MONITORIZAR REDES SDN

Clustering es la tarea de agrupar un conjunto de objetos de tal manera que los objetos en el mismo grupo (llamado cluster) son lo más similares (en cierto sentido) entre sí que con los de otros grupos. Es una tarea principal de la minería de datos exploratoria, y una técnica común para el análisis estadístico de datos, utilizado en muchos campos, incluido el aprendizaje automático, el reconocimiento de patrones, el análisis de imágenes, la recuperación de información, la bioinformática, la compresión de datos y los gráficos por ordenador.

El análisis de Clustering en sí no es un algoritmo específico, sino que es la tarea general a resolver. Se puede lograr mediante varios algoritmos que difieren significativamente en su comprensión de lo que constituye un cluster y cómo encontrarlos de manera eficiente. Las nociones populares de clusters incluyen clusters con pequeñas distancias entre los miembros del grupo, áreas densas del espacio de datos, intervalos o distribuciones estadísticas particulares entre otros. Por lo tanto, el término clustering puede formularse como un problema de optimización de objetivos múltiples. El algoritmo de clustering apropiado y la configuración de parámetros (incluidos los parámetros como la función de distancia a utilizar, un umbral de densidad o el número de clusters esperados) dependen del conjunto de datos individual y del uso previsto de los resultados. El análisis de conglomerados como tal no es una tarea automática, sino un proceso iterativo de descubrimiento de conocimiento u optimización interactiva de objetivos múltiples que implica pruebas y fallos. A menudo es necesario modificar el procesamiento previo de datos y los parámetros del modelo hasta que el resultado alcance las propiedades deseadas. Resumiendo, el objetivo de usar la agrupación es maximizar la variación inter-clusters y minimizar la variación intra-clusters.

Hemos utilizado dos algoritmos de agrupación en este trabajo:

- **Kmeans** [35]: es probablemente el algoritmo de clustering más conocido y el más simple entre los algoritmos de aprendizaje no supervisados. Este trabajo funciona según el principio de agrupación Kmeans. En realidad, esto significa que los grupos (clusters) para un conjunto de datos dado, están representados por una variable “k”. Para cada grupo, se define un centroide. El centroide es un punto de datos presente en el centro de cada grupo (considerando la distancia euclidiana).
- **Expectation-Maximization (EM)** [36]: es un método iterativo para buscar la máxima probabilidad o el máximo. Este algoritmo se basa en la distribución gaussiana en estadística. Considera una colección de distribuciones gaussianas para el conjunto de datos en un problema de LA.

Este trabajo es una extensión de la prueba de concepto utilizada en [9] donde las diferencias entre clustering (usando el algoritmo kmeans) y el algoritmo optimizado de conservación de flujo se pusieron de manifiesto usando la tasa de datos como métrica única en las simulaciones. Entonces, tomando como referencia los resultados de [9] (usando el algoritmo kmeans),

se han extendido las pruebas usando el algoritmo EM para comparar los resultados entre ambos algoritmos.

El presente trabajo se basa en la aplicabilidad y el estudio de los beneficios al aplicar diferentes algoritmos de clustering en la herramienta de monitorización presentada en [9]. Este estudio se verifica utilizando las métricas de tasa de datos y tasa de error, que también se describen en [9], para afirmar que es posible disminuir el número de peticiones de monitorización mientras se mantiene la precisión de los datos monitorizados.

Por tanto, el objetivo del presente trabajo es optimizar el número de solicitudes de switches utilizando diferentes algoritmos de clustering para reducir el número de solicitudes de switches respecto a técnicas no optimizadas entre las que se encuentran las herramientas de monitorización todos los switches de una topología. Es importante aclarar que los algoritmos de agrupación no se utilizan para optimizar el número de peticiones, ya que se utilizan para crear clusters sobre los cuales se ejecutan las técnicas de optimización descritas en [9]. Después de esto, se continúa aplicando diferentes algoritmos, obteniendo diferentes asignaciones de clusters y, por lo tanto, diferentes resultados en las simulaciones.

Antes de empezar a realizar peticiones de estadísticas a los switches, la función $Clustering_{Function}(G)$ lee la topología de red estructurada lógicamente en un grafo $G = (S, L)$ el cual será usado para construir los clusters. Debido a que G contiene el grafo de la red, la función $Create_Cluster(G, Algorithm, nc)$ los divide en N clusters, nombrándolos como $c_0, c_1, \dots, c_{nc-1}$.

La agrupación de los switches se basa en el número de puertos que están compuestos y la importancia de aplicar uno de los algoritmos implementados: Kmeans o Expectation-Maximization descritos en la sección anterior.

Una vez que se ha seleccionado el algoritmo, los criterios para monitorizar los clusters dependen del número de puertos de los switches. Por un lado, la herramienta de monitorización aplica el principio del algoritmo de conservación de flujo en switches de 2 grados, ya que todo el tráfico de datos que recibe el puerto entrante del switch se reenviará al puerto saliente del mismo switch y, por lo tanto, el tráfico a través de dichos enlaces será casi el mismo valor. Por otro lado, los clusters que contienen switches con tres o más puertos se eligen aleatoriamente en cada período T_{mon} que se supervisará. Para más nivel de detalle, estos algoritmos se describen en las secciones **4.2 Flow-Conservation Algorithm Enhancement** y **4.3 Clustering Enhancement Technique** de [9].

Para propósitos experimentales, hemos elegido la tasa de datos y la tasa de error como métricas para demostrar la viabilidad de nuestra propuesta. El procedimiento para calcular ambas métricas se describe en la sección **4.3 Clustering Enhancement Technique** de [9]. Concretamente, los procedimientos para calcular la tasa de datos y las métricas de error se explican en detalle en el Algoritmo 2 y el Algoritmo 4, respectivamente.

V. SIMULACIONES Y RESULTADOS

En esta sección, probamos los beneficios de los algoritmos de Clustering utilizando la herramienta de monitorización de optimización de Clustering descrita en la sección **4 Enhanced Monitored Algorithm** de [9].

Hemos realizado las pruebas en el escenario descrito en la Figura 1 atendiendo al uso de la optimización del clustering utilizando una plataforma de simulación llamada Mininet v2.1.0 [21]. La topología se compone de 7 OF-Switches ($s_1, s_2, s_3, s_4, s_5, s_6, s_7$) y 2 host (h_1 y h_2) conectados a s_1 y s_7 respectivamente. Los enlaces [$switch : s_3 - puerto : p_4, switch : s_6 - puerto : p_1$] y [$switch : s_6 - port : p_2, switch : s_5 - port : p_3$] están configurados con valores de velocidad de datos máxima (1 Mbps de ancho de banda) y porcentaje de pérdida del 5%.

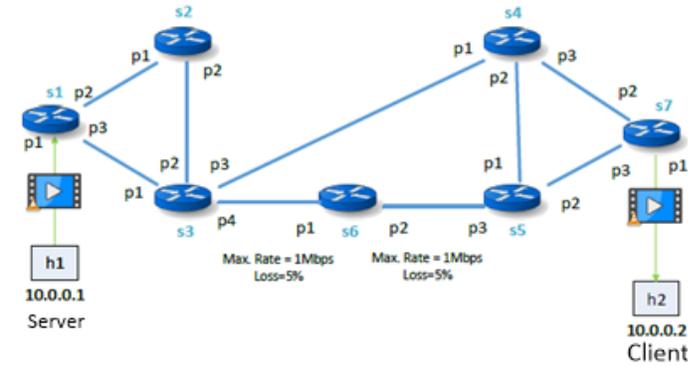


Figura 1. Topology tested.

Una simulación consiste en una transmisión de vídeo enviada desde el host 1 al host 2 usando el servidor de vídeo VLC y RTP/UDP como protocolo de transmisión al mismo tiempo que se ejecuta el módulo de monitorización. Por lo tanto, la tarea principal del módulo de monitorización es medir las estadísticas de los switches de la red (tanto los enlaces donde se envía el vídeo como el resto de ellos). El vídeo utilizado en la transmisión de vídeo es “Highway_cif” [38] y el tiempo de monitorización (t_{mon}) es de 200 ms.

La construcción de clusters será diferente atendiendo a la aplicación de los algoritmos de clúster Kmeans y EM como muestra la Tabla I. Una vez que se han creado los clusters, la herramienta de monitorización aplica las diferentes estrategias de monitorización descritas en [9].

Tabla I
CLUSTER ASSIGNMENT IN TEST SCENARIO

Clusters	Kmeans	EM
Cluster 0 (C_0)	S_3	$S_4S_5S_6$
Cluster 1 (C_1)	S_1S_6	S_1
Cluster 2 (C_2)	S_4S_5	S_2S_7
Cluster 3 (C_3)	S_2S_7	S_3

Como se ha indicado anteriormente, los resultados de las simulaciones aplicando el algoritmo kmeans se exponen en [9]. El número de consultas de monitorización en el método no optimizado durante la simulación (80 segundos) fueron 2079 solicitudes, mientras que en el algoritmo optimizado, la

aplicación del algoritmo de clustering Kmeans fue 903. Supuso una relación de reducción del 57 % en solicitudes con respecto a una simulación no optimizada.

Comenzando con los resultados de la simulación realizados en este trabajo y tomando la tasa de datos como la medición que aplica el algoritmo EM, el número de consultas de monitorización es 888. La diferencia de 1191 consultas (con el algoritmo no optimizado que fueron 2079 peticiones) muestra un relación de reducción del 58 %.

La Figura 2 muestra el flujo de tráfico entre dos simulaciones en las cuales una aplica la optimización usando el algoritmo EM respecto a otra que no lo aplica. La Figura describe el flujo de datos (en bps) a través de los enlaces $s_3 - s_6$. Las líneas puntiagudas (líneas azules) muestran la tasa de datos que el servidor está enviando, mientras que las líneas continuas (líneas verdes) muestran la tasa de datos obtenida utilizando el método no optimizado y las líneas de puntos (líneas rojas) muestran la velocidad de datos con algoritmo optimizado. Como se esperaba, los enlaces experimentan un aumento de la velocidad de datos debido a la transmisión del vídeo entre h1 y h2. Tan pronto como finaliza la transmisión (alrededor de 200 solicitudes), la tasa de datos disminuye, lo que demuestra la eficiencia del algoritmo para detectar cambios en la transmisión de la red. Estos resultados confirman que la optimización mantiene buenos niveles de precisión y reduce el número de solicitudes en el plano de datos utilizando dos criterios diferentes para construir el conjunto de switches. La diferencia entre el enlace $s_3 - s_6$ aplicando y sin aplicar la optimización es de 41,22 Kbps.

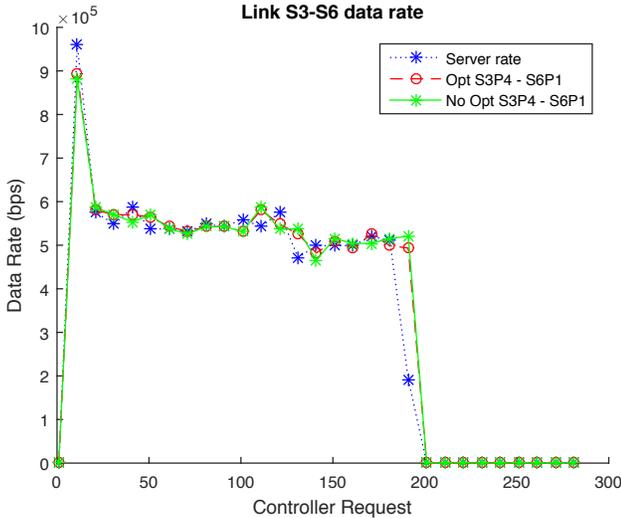


Figura 2. Data rate of link s_3-s_6

Por otro lado, la Figura 3 muestra la tasa de error en los datos de tráfico a través de la ruta de transmisión, es decir el enlace $s_3 - s_6$ (*switch* : $s_3 - puerto$: p_4 , *switch* : $s_6 - puerto$: p_1). Las líneas continuas (líneas verdes) muestran la tasa de error obtenida utilizando el método no optimizado, las líneas punteadas (líneas rojas) muestran la tasa de error con el algoritmo optimizado y la línea continua punteada (líneas azules) muestra el porcentaje netem configurado en Mininet

(5 % de pérdida de paquetes). Como se esperaba, los enlaces pueden detectar la pérdida de información entre los enlaces debido a las características de estos enlaces.

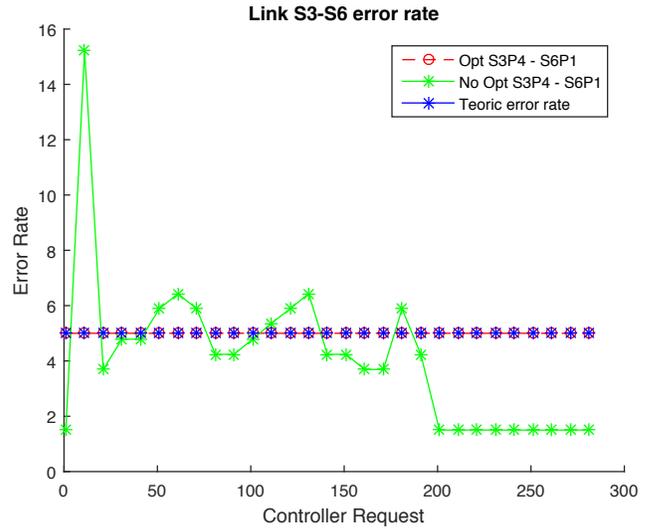


Figura 3. Error rate of link s_3-s_6

VI. CONCLUSIONES

SDN ha traído muchas oportunidades en el ámbito de las redes que han permitido la adaptación dinámica y la reconfiguración con su separación del plano de control del dispositivo de reenvío controlado por el controlador centralizado. El controlador SDN gestiona los flujos de red de forma dinámica e individual con la visión global de toda la red. El surgimiento de SDN con su capacidad de control dinámica puede traer varios beneficios a centros de datos en la nube donde, en esencia, los requisitos y la utilización cambia dinámicamente bajo demanda. Hay una amplia gama de algoritmos de clustering, cada uno con sus ventajas e inconvenientes con respecto a qué tipo de datos agrupan, complejidad de tiempo, debilidades, etc. Existen otros algoritmos de clustering entre los que se encuentran por ejemplo clustering aglomerativo jerárquico (o clustering de vinculación), bueno para cuando no haya necesariamente grupos circulares (o hiper esféricos), y no conozca la cantidad de grupos por adelantado. Comienza con cada punto siendo un cluster separado y funciona uniendo los dos grupos más cercanos en cada paso hasta que todo esté en un grupo grande. Este artículo presenta un estudio de los beneficios aplicando diferentes algoritmos de clustering en una herramienta de monitorización que reduce la cantidad de solicitudes de monitorización en los dispositivos que la red que está compuesto. Esta reducción en las consultas a los switches de la red depende de la construcción de los clusters. Hemos demostrado que usando diferentes algoritmos de clustering para construir diferentes conjuntos/grupos de switches, proporciona beneficios disminuyendo el número de consultas de monitorización en los switches de la topología. Usando los valores de tasa de datos y tasa de error obtenidos a partir de los resultados de las simulaciones, se confirman

los beneficios de los algoritmos de clustering respecto a las simulaciones no optimizadas.

ACKNOWLEDGMENT

This paper has also received funding from THEIA (Techniques for Integrity and authentication of multimedia files of mobile devices) UCM project (FEI-EU-19-04).

REFERENCIAS

- [1] M. T. Seufert, "Quality of Experience and Access Network Traffic Management of HTTP Adaptive Video Streaming", Doctoral Thesis, University of Würzburg, 2017. [Online]. Available: https://opus.bibliothek.uni-wuerzburg.de/files/15413/Seufert_Michael_Thomas_HTTP.pdf
- [2] P. Casas, M. Seufert, and R. Schatz, "YOUQMON: A System for Online Monitoring of YouTube QoE in Operational 3G Networks," ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 2, pp. 44–46, 2013.
- [3] M. Seufert, M. Slanina, S. Egger, and M. Kottkamp, "To Pool or not to Pool: A Comparison of Temporal Pooling Methods for HTTP Adaptive Video Streaming," in Proceedings of the 5th International Workshop on Quality of Multimedia Experience (QoMEX), Klagenfurt, Austria, 2013.
- [4] M. Wennesheimer, D. Robinson, J. De Vriendt, D. De Vleeschauwer, C. Bahr, K. Heise, S. Argyropoulos, B. Feiten, M. Bilgic, T. Raiha, S. Egger, M. Seufert, M. Slanina, M. Kottkamp, H. Gsödl, H. Ibl, and R. García Pérez, "Service Quality Definition and Measurement," Next Generation Mobile Networks Alliance (NGMN), Tech. Rep., 2013, version 1.0.4
- [5] F. Wamser, M. Seufert, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "YoMoApp: a Tool for Analyzing QoE of YouTube HTTP Adaptive Streaming in Mobile Networks," in Proceedings of the European Conference on Networks and Communications (EuCNC), Paris, France, 2015.
- [6] M. Seufert, F. Wamser, P. Casas, R. Irmer, P. Tran-Gia, and R. Schatz, "YouTube QoE on Mobile Devices: Subjective Analysis of Classical vs. Adaptive Video Streaming," in Proceedings of the 6th International Workshop on Traffic Analysis and Characterization (TRAC), Dubrovnik, Croatia, 2015.
- [7] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, V. Maglaris, Combining OpenFlow and sFlow for an Effective and Scalable Anomaly Detection and Mitigation Mechanism on SDN Environments, Computer Networks 62 (1) (2013) 122–136.
- [8] A. Zaalouk, R. Khondoker, R. Marx, K. M. Bayarou, Orchsec: An orchestrator-based architecture for enhancing network-security using network monitoring and sdn control functions., in: NOMS, 2014, pp. 1–9.
- [9] Puente Fernández, J., García Villalba, L., and Kim, T. H. (2018). Clustering and Flow Conservation Monitoring Tool for Software Defined Networks. Sensors, 18(4), 1079.
- [10] McKeown N.; Anderson T.; Balakrishnan H.; Parulkar G.; Peterson L.; Rexford J.; Shenker S.; Turner J. OpenFlow: Enabling Innovation in Campus Networks. ACM SIGCOMM Computer; Communication Review; 2008; vol. 38 (2); pp. 69–74.
- [11] OpenFlow Switch Specification v1.1.0. Available at <http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf>
- [12] Gude N.; Koponen T.; Pettit J.; Pfaff B.; Casado M.; McKeown N.; Shenker S. NOX: Towards an Operating System for Networks. ACM SIGCOMM Computer Communication Review; 2008; vol. 38; pp. 105–110.
- [13] Project Floodlight: Open Source Software for Building Software-Defined Networks. Available at <http://www.projectfloodlight.org>
- [14] OpenDaylight (ODL) open source SDN platform. Available at <https://www.opendaylight.org>
- [15] Foster, N.; Harrison R.; Monsanto C.; Rexford J.; Story A.; Walker, D. Frenetic: A Network Programming Language. In ACM SIGPLAN Notices; 2011; vol. 46(9); pp. 279–291.
- [16] Voellmy A.; Kim H.; Feamster N. ProCera: a Language for High-level Reactive Network Control. In Proceedings of the first workshop on Hot topics in software defined networks; ACM; 2012; pp. 43–48.
- [17] Monsanto C.; Foster N.; Harrison R.; Walker D. A Compiler and Runtime System for Network Programming Languages. In ACM SIGPLAN Notices; ACM; 2012; vol 47(1); pp. 217–230.
- [18] Voellmy A.; Wang J. Scalable Software Defined Network Controllers. In Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication; ACM; 2012; pp. 289–290.
- [19] M. Casado, N. Foster, A. Guha, Abstractions for software-defined networks, Communications of the ACM; 2014; pp. 86–95.
- [20] N. Feamster, J. Rexford, E. Zegura, The Road to SDN: An Intellectual, History of Programmable Networks, ACM SIGCOMM Computer Communication Review 44 (2) (2014) 87–98.
- [21] V. K. Gurbani, M. Scharf, T. Lakshman, V. Hilt, E. Marocco, Abstracting network state in software defined networks (sdn) for rendezvous services, in: 2012 IEEE International Conference on Communications (ICC), IEEE, 2012, pp. 6627–6632
- [22] Case J. D.; Fedor M.; Schoffstall M. L.; Davin J. Simple Network Management Protocol (SNMP). RFC 1157 (Historic); 1990.
- [23] Enns R.; Bjorklund M.; Schoenwaelder J.; Bierman A. Network Configuration Protocol.
- [24] Claise B. RFC 3954 - Cisco Systems NetFlow Services Export Version 9. RFC 3954; 2004.
- [25] Phaal P.; Lavine M. Sflow version 5. 2004.
- [26] Myers A. C. JFlow: Practical Mostly-static Information Flow Control. In Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages; ACM; 1999; pp. 228–241.
- [27] Gosling, J., Joy, B., Steele, G. L., Bracha, G., and Buckley, A. (2014). The Java Language Specification. Pearson Education.
- [28] Shibuya M.; Tachibana A.; Hasegawa T. Efficient Performance Diagnosis in OpenFlow Networks Based on Active Measurements. 1st ACM Conference on Information-Centric Networking (ICN-2014); 2014; pp. 279.
- [29] Valdivieso Caraguay A. L.; Puente Fernández J. A.; García Villalba L. J. An Optimization Framework for Monitoring of SDN/OpenFlow Networks. International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC); 2015.
- [30] Van Adrichem N. L.; Doerr C.; Kuipers F. A. Opennetmon: Network Monitoring in Openflow Software-defined Networks. In Network Operations and Management Symposium (NOMS); IEEE; 2014; pp. 1–8.
- [31] Tootoonchian A.; Ghobadi M.; Ganjali Y. OpenTM: Traffic Matrix Estimator for OpenFlow Networks. In Passive and active measurement; 2010; pp. 201–210.
- [32] Chowdhury, S. R., Bari, M. F., Ahmed, R., and Boutaba, R. (2014, May). Payless: A low cost network monitoring framework for software defined networks. In 2014 IEEE Network Operations and Management Symposium (NOMS) (pp. 1-9). IEEE.
- [33] Yu, C., Lumezanu, C., Zhang, Y., Singh, V., Jiang, G., Madhyastha, H. V. (2013, March). Flowsense: Monitoring network utilization with zero measurement cost. In International Conference on Passive and Active Network Measurement (pp. 31-41). Springer Berlin Heidelberg.
- [34] J. Suárez-Varela, P. Barlet-Ros, Sbar: Sdn flow-based monitoring and application recognition, in: Proceedings of the Symposium on SDN Research, ACM, 2018, p. 22
- [35] HARTIGAN, John A.; WONG, Manchek A. Algorithm AS 136: A k-means clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 1979, vol. 28, no 1, p. 100-108.
- [36] MOON, Todd K. The expectation-maximization algorithm. IEEE Signal processing magazine, 1996, vol. 13, no 6, p. 47-60.
- [37] Mininet. Available at <https://github.com/mininet/mininet/wiki/Introduction-to-Mininet>
- [38] Highway. Available at <http://www2.tkn.tu-berlin.de/research/evalvid/qcif.html>

Jesús Antonio Puente Fernández was born in Madrid (Spain) in 1988. Jesús received his Computer Science Engineering degree by Universidad Complutense de Madrid (Spain) in 2012. He received a M.S. degree in Computer Research in the Universidad Complutense of Madrid (Spain) in 2014. He is currently a Ph.D. student of Computer Engineering in Universidad Complutense de Madrid (Spain). His research interests include computer networks, software-defined networking and network function virtualization. In addition, he is currently working as IT security evaluator and Project manager in DEKRA Testing and Certification.

Luis Javier García Villalba received a Telecommunication Engineering degree from the Universidad de Málaga (Spain) in 1993 and holds a Ph.D. in Computer Science (1999) from the Universidad Politécnica de Madrid (Spain). Visiting Scholar at COSIC (Computer Security and Industrial Cryptography, Department of Electrical Engineering, Faculty of Engineering, Katholieke Universiteit Leuven, Belgium) in 2000 and Visiting Scientist at IBM Research Division (IBM Almaden Research Center, San Jose, CA, USA) in 2001 and 2002, he is currently Associate Professor of the Department of Software Engineering and Artificial Intelligence at the Universidad Complutense de Madrid (UCM) and Head of Complutense Research Group GASS (Group of Analysis, Security and Systems) which is located in the Faculty of Computer Science and Engineering at the UCM Campus. His professional experience

includes the management of both national and international research projects and both public (Spanish Ministry of R&D, Spanish Ministry of Defence, Horizon 2020 - European Commission, . . .) and private financing (Hitachi, IBM, Nokia, Safelayer Secure Communications, TB Solutions Security, . . .). Author or co-author of numerous international publications is editor or guest editor of numerous journals such as Entropy MDPI, Future Generation Computer Systems (FGCS), Future Internet MDPI, IEEE Latin America Transactions, IET Communications (IET-COM), IET Networks (IET-NET), IET Wireless Sensor Systems (IET-WSS), International Journal of Ad Hoc and Ubiquitous Computing (IJAHUC), International Journal of Multimedia and Ubiquitous Engineering (IJMUE), Journal of Supercomputing, Sensors MDPI, etc.