ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE SISTEMAS

UNIDAD DE TITULACIÓN

Análisis del Impacto de un Ataque DoS en la Calidad de Servicio de Sistemas Streaming Multimedia en Redes SDN

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN MAGISTER EN SOFTWARE

CHRISTIAN GIOVANNY MANTILLA YUNGA
cristian_jens@hotmail.com

Director: DR. ÁNGEL LEONARDO VALDIVIESO CARAGUAY
angel.valdivieso@epn.edu.ec

2021

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación “Análisis del Impacto de Ataque DoS en la Calidad de Servicio de Servicios Streaming Multimedia en Redes SDN”, desarrollado por Christian Giovanny Mantilla Yunga estudiante de la Maestría en Software, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.

ANGEL
LEONARDO
VALDIVIESO
CARAGUAY

Firmado digitalmente por ANGEL LEONARDO VALDIVIESO CARAGUAY
Nombre de reconocimiento (DN): cn=ANGEL LEONARDO VALDIVIESO CARAGUAY, serialNumber=020221094550, ou=ENTIDAD DE CERTIFICACION DE INFORMACION, o=SECURITY DATA S.A. 2, c=EC
Fecha: 2021.06.01 10:44:35 -05'00'

Dr. Ángel Leonardo Valdivieso Caraguay

DIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, Christian Giovanny Mantilla Yunga declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Christian Giovanny Mantilla Yunga

DEDICATORIA

A la memoria de mi Padre fallecido Luis Mantilla quién ha sido un referente en la educación y en la cual me encaminado. A mi Madre Elizabeth Yunga quién fue pilar fundamental para mi educación y un ejemplo a seguir de su dedicación al trabajo duro para conseguir sus metas, en el cual me veo reflejado para conseguir las mías. Mi hermana Anahí Romero el cual me ha brindado su apoyo cada día y es una parte importante en mi vida. A mis segundas Madres Mercedes Mora y Narcisa Yunga quienes guiaron mi camino desde pequeño, compartiendo cada segundo de mis estudios y vida personal. Por último, a toda mi Familia Yunga que me han apoyado en diferentes aspectos de mi vida, todos mis éxitos en adelante se los dedico a ellos.

Christian Mantilla

AGRADECIMIENTO

Mis más sinceros agradecimientos a mi tutor Leonardo Valdivieso, por haber hecho posible la obtención de mi título de Máster en software y por todo el apoyo en el camino de la tesis y el artículo profesional, un gran profesor y profesional.

A Lorena Barona y Pamela Flores, dos excelentes profesoras que sin su apoyo no podría haber terminado el grado de Máster, les quedo eternamente agradecido.

A Sang Guun y Roberto Andrade por haberme brindado un espacio en su servidor para que pueda realizar el trabajo de tesis, muchísimas gracias.

A mis grandes amigos que a la final son como mis hermanos José Luis Escobar, Luis Castillo y Mauricio Cabrera, quienes fueron incondicionales, brindándome apoyo para continuar con la carrera de posgrado, lo logramos.

A mis Amigos Christian Enríquez, Marcela Mosquera, Wendy Parra, que siempre estuvieron atrás mío para que siga con la tesis.

Un agradecimiento a Carolina Cáceres quién me ha enseñado a lo largo de la carrera de posgrado a ser perseverante y nunca dejarse caer ante las adversidades

.

Christian Mantilla

ÍNDICE DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

RESUMEN

Las Redes Definidas por Software (SDN) se destacan por la flexibilidad para adaptarse a nuevos servicios y reducir costos de gestión. Por otra parte, las aplicaciones de streaming multimedia son uno de los servicios más importantes en la red, aunque exigen altos requerimientos sobre la calidad de servicio (QoS). A pesar de los beneficios que ofrece SDN sobre servicios de streaming multimedia, existen peligros de vulnerabilidad que pueden ocasionar varios ataques sobre la red, tal es el caso de ataques de Denegación de Servicio (DoS). Los efectos de un ataque DoS en redes SDN sobre la calidad de un servicio multimedia no han sido analizados a profundidad. Por este motivo, el presente trabajo propone una arquitectura de análisis y pruebas de QoS/QoE que permite efectuar diferentes tipos de ataque DoS sobre los elementos de una red SDN que transmite información multimedia en tiempo real. La implementación de la arquitectura permite comparar un escenario normal vs. un escenario bajo ataque DoS y su efecto en la QoS/QoE en términos de Detail Loss Metric/ADM2 (DLM/ADM2), Structural Similarity Index Measure (SSIM), Peak Signal to Noise Ratio (PSNR) y Video Multimethod Assessment Fusion (VMAF). Los ataques de prueba realizados son: spoofing a la red SDN, flooding al servidor de streaming multimedia y al servidor cliente. Los resultados obtenidos demuestran que el ataque al cliente multimedia tiene mayor efectividad, tal es el caso de métricas QoS como: ADM2 que se reduce en un 20%, PSNR en un 56%, SSIM en un 24%, y para QoE con VMAF, se obtiene un puntaje de 54% de percepción de calidad de video.

Palabras clave: {Redes Definidas por Software, Video Streaming, QoS/QoE, VMAF; DoS}

ABSTRACT

Software Defined Networks (SDN) stand out for their flexibility to adapt to new services and reduce management costs. Additionally, multimedia streaming applications are one of the most important services in the network, although they demand high requirements on the quality of service (QoS). Despite the benefits that SDN offers over multimedia streaming services, vulnerabilities can use can cause various attacks on the network, such as Denial of Service (DoS) attacks. The effects of a DoS attack on the QoS of multimedia services in SDN networks have not been thoroughly analyzed. For this reason, this work proposes a proposes an arquitecture to test and analyze the effect of different types of DoS attacks to be carried out on the elements of an SDN network that transmit multimedia information in real time. The architecture enables the comparison between two scenarios (normal vs. under attack) and the effect on QoS / QoE in terms of Detail Loss Metric / ADM2 (DLM / ADM2), Structural Similarity Index Measure (SSIM), Peak Signal to Noise Ratio (PSNR) and Video Multimethod Assessment Fusion (VMAF). The test attacks carried out are: spoofing to the SDN network, flooding to the multimedia streaming server and the client server. The results obtained show that the attack on the multimedia client is more effective: in QoS, the ADM2 which is reduced by 20 %, PSNR by 56 %, SSIM by 24 %, and for QoE with VMAF, you get a score of 54 \% video quality perception.

Keywords: {Software Defined Networking, SDN, Streaming Video, QoS/QoE, VMAF, DoS Attack}

1. INTRODUCCIÓN

Las Redes Definidas por Software (SDN) constituyen una arquitectura emergente que ayuda a mitigar problemas de infraestructura física difíciles de solucionar, como son la rápida adaptabilidad a nuevos servicios, implementación de soluciones específicas (firewall, switches, balanceadores) utilizando un mismo hardware, entre otros. Esta arquitectura rompe el esquema tradicional teniendo como resultado una fácil administración y bajo costo de implementación (Liberato et al., 2018). SDN promueve la centralización lógica del control de la red y a su vez separa la arquitectura del hardware (plano de datos) con respecto al software de ejecución de las aplicaciones de red (plano de control) (Schaller & Hood, 2017). De este modo, permite que la administración de servicios de la infraestructura de red sea fácilmente programable acorde a las necesidades particulares de la organización (Karakus & Durresi, 2017a). El uso de SDN se ha extendido ampliamente en múltiples campos de aplicación; por ejemplo, en sistemas IoT, SDN permite mejorar el control y la gestión de dispositivos (Caraguay et al., 2018). De igual manera, en redes 5G el uso de SDN mejora el desempeño de aplicaciones en tiempo real (Gabriel et al., 2018) entre otros. Uno de los campos relevantes en los cuales SDN provee ventajas importantes son los servicios multimedia. La demanda de contenido multimedia en los sectores de comunicación, entretenimiento y educación han aumentado ampliamente en los últimos años (Kunwar Pal et.al., 2019). El 90% del tráfico de red que existe en Internet es producido por servicios multimedia basados en video streaming (Stefano Petrangeli at el., 2016). En este contexto, el uso de sistemas SDN evita el congelamiento de video cuando se tiene bajos recursos de red (Danda B. Rawat & Reddy, 2017), (Bentaleb et al., 2017), (Zhao & Medhi, 2017).

En este sentido, la Figura 1 muestra el tráfico de video sobre internet desde el año 2016 y una proyección hasta el año 2021. Por su parte, la Figura 2 realiza la misma proyección de video streaming, subcategorizado en Video on Demand (VoD) y Streaming Video. En primer lugar, VoD se refiere al acceso a un contenido en concreto, no existe restricción de tiempo de transmisión, solo el tiempo que dura el video. En segundo lugar, Video Streaming permite que el contenido sea observado mientras dure el tiempo de transmisión (Stefano Petrangeli et al , 2016) (Kunwar Pal, 2016). Ejemplos de cores de negocio basados en estos paradigmas incluyen Netflix, Youtube en (VoD), Microsoft Smooth Streaming y Apple HTTP Live Streaming (HLS) en Video Streaming (Stefano Petrangeli et al, 2016) (Kunwar Pal, 2016). En la Figura 1 se muestra que la proyección del tráfico de Internet alcance un 67.4%, para el 2021, al igual que se mantiene durante los años 2016-2021 por encima de los demás servicios que ofrece el internet. Sin embargo, en la Figura 2 video streaming

ocupa un segundo lugar dentro de la subcategorización, aún así, el contenido que se ofrece para video streaming ha venido en aumento.

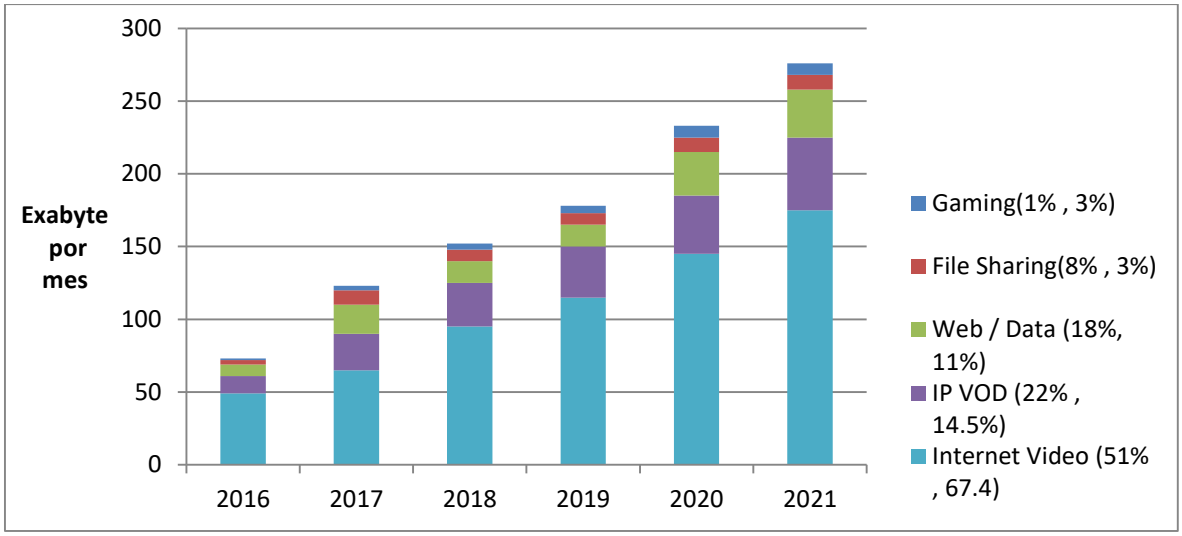

Figura. 1 -Tráfico sobre internet (Kunwar Pal, 2016).

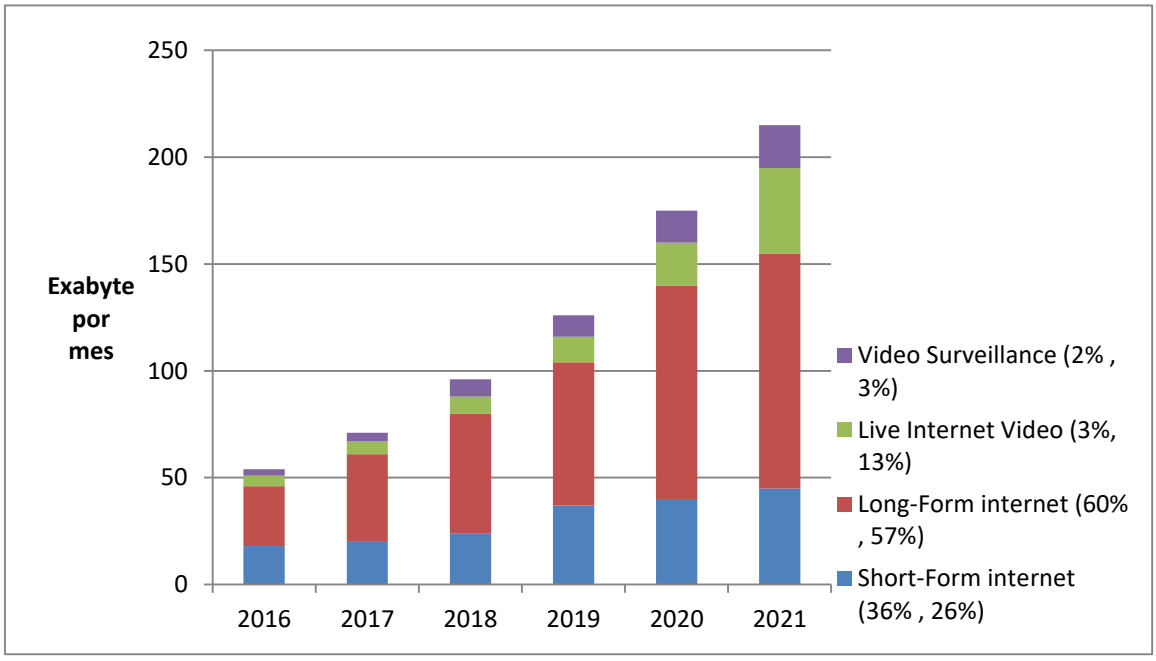

Figura. 2 -Tráfico en internet - Clasificación de video (Kunwar Pal, 2016)

Por otra parte, el acceso a la información on-line ha tenido una gran evolución en los últimos años. Cientos de empresas apuestan a que gran parte de sus transacciones y servicios se realicen por la web (Sachdeva et al., 2010). Esto genera un riesgo de seguridad ya que los procesos, transacciones bancarias o servicios multimedia pueden ser interrumpidos o

verse vulnerados mediante técnicas de ataques como: man in the middle, malware, ransomware, Denegación de Servicio (DoS), entre otros. En este sentido, uno de los principales ataques recibidos son los ocasionados por DoS, el mismo que consiste en el envío agresivo de tráfico a la red (Oktian et al., 2014). En los últimos años se han realizado investigaciones que buscan analizar y mitigar ataques DoS sobre una red SDN (Hyun et al., 2017)(Douligeris & Mitrokotsa, 2004) (Guofei Gu, 2008).

Con estos antecedentes, el presente artículo analiza del comportamiento de aplicaciones de servicios streaming multimedia sobre redes SDN; en especial en escenarios bajo ataques DoS como una de las principales amenazas existentes. En resumen, se propone una arquitectura que permita analizar el efecto de un ataque DoS en la calidad de servicio de sistemas streaming multimedia en redes SDN. La arquitectura será implementada utilizando herramientas Open Source y se realizarán pruebas en una Topología de Red tipo Data Center. Los resultados serán analizados mediante la medición de métricas objetivas (QoS), como por ejemplo: PSNR, SSIM, DLM/ADM2, y subjetivas (QoE), como VMAF. Estos resultados se contrastarán con un escenario ideal; es decir, sin ataques e interrupciones en la red SDN. Los resultados permitirán mejorar el diseño de nuevos algoritmos y estrategias de mitigación de este tipo de ataques.

1.1. Pregunta de investigación

¿Existe una arquitectura que analice el Impacto de Ataque DoS en la Calidad de Servicio de Servicios Streaming Multimedia en Redes SDN?

1.2. Objetivo general

Analizar el impacto en la calidad de servicio de un ataque DoS en un sistema streaming de video sobre redes SDN.

1.3. Objetivos específicos

- Revisar el estado del arte (SDN, DoS, streaming multimedia, QoS) y seleccionar las herramientas que faciliten el análisis de la QoS, generar un ataque DoS, así como emular una red SDN.
- Definir un esquema que permita evaluar el efecto de un ataque DoS en un servicio multimedia de streaming para una red tipo SDN.

- Diseñar los experimentos de ataques DoS en servicios streaming multimedia para redes SDN.
- Implementar el esquema propuesto y ejecutar los experimentos.
- Recolectar y Analizar los resultados obtenidos.
- Evaluar y documentar los resultados en un artículo científico.

2. METODOLOGÍA

Este proyecto tendrá dos escenarios de pruebas que serán evaluados y analizados. Es por ello que es imprescindible contar con una metodología de investigación acorde al tema propuesto. Entre las principales metodologías se tiene:

- **No experimental:** no hay manipulación de variables.
- **Experimentales:** permite modificar y controlar variables
- **Cuasi-experimentales:** no tiene tal grado de manipulación, ya que a los elementos que se investigan se les pre-asignan condiciones particulares.
- **Pre – experimental:** no hay selección aleatoria de los elementos ni se incluye un grupo de control (Tamayo y Tamayo, 1980).

Dichas metodologías conceptualmente tienen semejanzas entre ellas, sin embargo, su diferentica radica en la manipulación de las variables junto al grupo de control (Tamayo y Tamayo, 1980). Dicho esto, se hace necesario el uso de la metodología experimental, que según Alberto Ramírez en el libro de Metodología e Investigación la define como: “La que permite establecer causación o relación de causa y efecto de un fenómeno a través de procedimientos controlados donde se manipulan y se controlan las variables que ejercen incidencia sobre el fenómeno. Por tanto, se tiene el control sobre el tratamiento en estudio” (Tamayo y Tamayo, 1980). Este método es aplicado para los casos de pruebas que permiten el control de variables que deben ser definidas, al igual que modificadas, observadas y posteriormente ser analizadas (Tamayo y Tamayo, 1980). Las variables dependientes e independientes se describen en la Tabla 1.

Tabla 1 - Variables Dependientes e Independientes Metodología Experimental.

Variables	**Descripción**
Variable independiente. - Es la variable que tiende a ser manipulada en el experimento.	La variable a modificarse es un sistema streaming multimedia en el cual se efectuará un ataque de denegación de servicios al controlador de SDN. El ataque ocasionará un aumento en la tasa de transmisión de los paquetes hacia otros elementos de red o al controlador SDN.

Variable dependiente. - Característica o propiedad que mediante la manipulación trata de cambiar la característica de la variable independiente.	Impacto que tiene un ataque de denegación de servicios sobre sistema streaming multimedia en una red SDN. El ataque afectará el QoS y QoE del servicio streaming y su impacto será medido en términos de ADM2. PSNR, SSIM, VMAF.

La razón principal del uso del método experimental es que se realizará pruebas controladas que tendrán dos tipos de grupos establecidos. Además, esta metodología nos permitirá entender el fenómeno que se produce en cada uno de ellos y posteriormente analizar los resultados obtenidos.

Este proyecto estará definido en 3 fases: la primera fase se realizará investigación del estudio del arte de DoS, SDN, multimedia streaming y la selección de una herramienta para medir QoS. En la segunda fase se definirá un esquema de trabajo para evaluar la QoS de transmisión de video. Luego se implementará y ejecutará el esquema de evaluación; la ejecución estará compuesta de dos tipos de grupo de pruebas. El primer grupo incluye el funcionamiento normal del servicio y el segundo el funcionamiento cuando el servicio está sufriendo un ataque DoS. Por último, en la tercera fase se recolectará, analizará y evaluará los resultados que posteriormente serán documentados para el proyecto.

3. MARCO TEÓRICO

En este capítulo se analizarán los conceptos principales usados en el proyecto.

3.1. Redes Definidas por Software (SDN)

Las redes SDN separan el plano de datos y plano de control de los dispositivos de red y centralizan la toma de decisiones a un solo servidor (controlador). La arquitectura SDN está definida por 3 capas (Danda B. Rawat & Reddy, 2017): capa de infraestructura, capa de control y capa de aplicación, tal como se muestra en la en la Figura 3

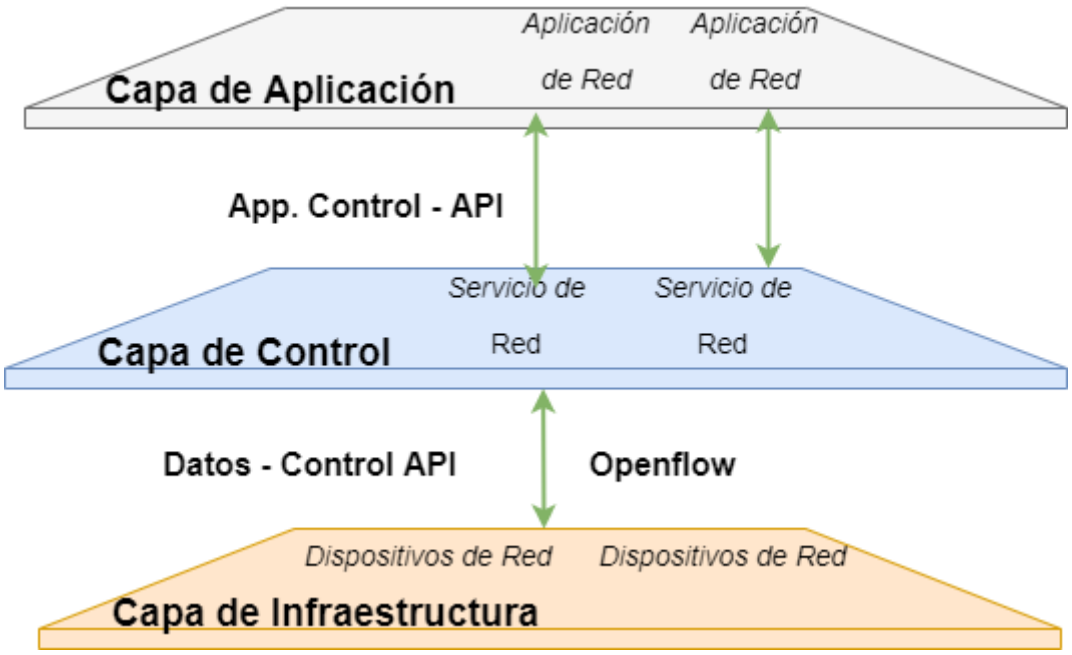

Figura. 3 - Arquitectura de una red SDN (Danda B. Rawat & Reddy, 2017)

- **Capa de infraestructura:** Está compuesta por dispositivos de red (hardware), quienes reciben instrucciones del plano de control. Se encarga de la recepción, lectura de cabeceras y re-envío de paquetes según las instrucciones de capas superiores.
- **Capa de control:** Controla las funciones básicas de red, supervisa el enrutamiento y administra el flujo de tráfico. Para la comunicación con la capa de infraestructura se utilizan interfaces (API) como: Openflow, Netconf, etc.
- **Capa de Aplicación:** Se implementa aplicaciones de alto nivel tanto comerciales, optimización, seguridad, entre otros, tales como: sistemas de optimización de red, detección de amenazas, balanceo de carga, control de acceso, entre otras. La comunicación con la capa de control se lo realiza mediante la interfaz App - Control.

SDN permite que la gestión de redes sea más eficiente, mejorando el rendimiento de la red a través del uso eficiente de los recursos y disminución del consumo de energía (Danda B. Rawat & Reddy, 2017). Dentro de las aplicaciones de SDN se puede citar: la sinergia entre tecnología celular y SDN, que puede ofrecer una mayor velocidad de transmisión de datos, debido al control del flujo de enrutamiento dinámico en la capa de control, y una mejora QoS a los usuarios. Así mismo, en redes Wifi se posibilita el uso efectivo de tiempo compartido en segmentos de red (Danda B. Rawat & Reddy, 2017).

3.2. Calidad de Servicio (QoS) y Calidad de Experiencia (QoE)

Al hablar de calidad en streaming multimedia en una red SDN hay que tomar en cuenta dos conceptos importantes y son: Calidad de Servicio y Calidad de Experiencia de Usuario.

3.2.1. Calidad de servicio (QoS)

QoS está muy ligado a los parámetros de capacidad de red como, enlace de ancho de banda, delay, pérdida de paquetes, entre otros (Ab & Asmi, 2017). Así mismo, en (Karakus & Durresi, 2017b) definen a QoS como la capacidad de tráfico de red necesaria para proporcionar servicios de alta calidad, es decir los parámetros de QoS son priorizados para alcanzar la máxima capacidad de red. Esta métrica tiene varios aspectos que pueden ser evaluados, como la tasa de pérdida de paquetes, delay y variación del delay. Para evaluar una adecuada QoS existen varias arquitecturas y frameworks que buscan mejorar la eficiencia y costos para proveedores. El cálculo de QoS en videos streaming, especialmente para codificaciones H264/ MPEG-4, tiene más dificultad debido a que contienen uno o dos subconjuntos de capas de flujos de bits. De tal manera, que la primera capa de flujo de HPEG-4 se codifica en una capa base en la que debe transmitirse sin pérdida de paquetes, ni variaciones. Por último, la segunda capa de mejora enfocado a QoS, se considera como flujos de mejor esfuerzo (Yu et al., 2015).

Por otra parte en (Egilmez & Tekalp, 2014), se describe que el Grupo de Trabajo de Ingeniería (IETF) presenta dos arquitecturas para el tráfico multimedia y son: QoS hard y QoS soft. El QoS hard garantiza la calidad de las conexiones y requisitos de QoS, pero por lado no tiene en cuenta las limitaciones de recursos, ejemplo método IntServ. Mientras

tanto que QoS soft no es estricto con los requisitos de QoS, un ejemplo de este método es DiffServ.

3.2.2. Métricas Objetivas

Para la percepción de video en QoS también se ha desarrollado numerosas métricas para evaluar estos parámetros (Vranješ et al., 2013). Estos métodos objetivos consisten en algoritmos y fórmulas para evaluar la calidad. Según la cantidad de información para comparar con el video original se lo puede categorizar en 3:

- **Referencia Completa:** El video original y el video después del streaming están disponibles.
- **Sin Referencia:** Solo el video recibido está disponible.
- **Referencia Reducida:** El video recibido y disponibilidad de características del video original también están disponibles (Cherif, 2013).

Las Métricas más conocidas en el cálculo de QoS son: Structural Similarity Index Measure (SSIM), Peak Signal to Noise Ratio (PSNR).

PSNR es una métrica objetiva que mide la relación de señal pico a ruido, por lo que se mide en decibelios. Su fórmula matemática se representa de la siguiente manera:

$$\mathbf{PSNR} = \mathbf{20} * \mathbf{log_{10}} \left(\frac{\boldsymbol{L^2}}{\boldsymbol{MSE}}\right)$$

Donde MSE es el error cuadrático medio, pero se lo representa como RMSE que es la raíz cuadrada. El MSE se calcula los pixeles del video recibido con respecto al video original. Es decir que para obtener el resultado de PSNR se necesita una referencia completa. En la ecuación “L” representa el rango de pixeles (Rassool, 2017).

Mientras que **SSIM**, evalúa la distorsión de dos imágenes en base a los valores luminancia, contraste y textura, estos valores se calculan para cada pixel individual, para una puntuación global los valores son promediados y están con respecto a una imagen de referencia. Sus valores oscilan entre 0-1 y se expresan en una escala de decibelios no lineales, su fórmula es:

$$\mathbf{SSIM}(\mathbf{x}, \mathbf{y}) = [l(x,y)]^{\alpha}[l(x,y)]^{\beta}[l(x,y)]^{\gamma}$$

Cabe mencionar que SSIM también es una métrica de referencia completa (Rassool, 2017).

3.2.3. Calidad de Experiencia de Usuario (QoE)

La calidad de experiencia de usuario es la manera correcta donde los proveedores de video pueden cuantificar sobre un puntaje la calidad de video, se trata de la percepción obtenida por el usuario. En (J. Li et al., 2018) define a QoE como el parámetro que refleja la satisfacción del observador y el cumplimiento de las expectativas del contenido del video. La QoE puede verse influida por varios factores; por lo que en el documento técnico de Qualinet (Brunnström et al., 2013), define estos factores en factor de sistema, factor de contexto y factor humano .Algunos proveedores de red tienden a enfocarse más en la QoE que en la QoS (Ab & Asmi, 2017).

3.2.4. Métricas Subjetivas

Las métricas subjetivas pueden obtenerse de encuestas a usuarios finales o de modelos empíricos establecidos que vinculan diferentes medidas de rendimiento, como retraso, calidad de imagen. Estos parámetros están dados en forma de puntuaciones medias de opinión o MOS (Liotou et al., 2018). Debido a que las pruebas subjetivas se realizan con el usuario final se deben cumplir ciertos parámetros estandarizados, es así que se puede usar las recomendaciones de la UIT- T P.910, que servirá para estandarizar el entorno de visualización. Además, se implementa la metodología Absolute Category Rating (ACR) como método de calificación, ya que permite una gran cantidad de condiciones de pruebas en una sola sesión (Lee et al., 2018). Actualmente, la métrica QoE más utilizada en la comunidad científica es Video Multimethod Assessment Fusion (VMAF).

VMAF es un algoritmo desarrollado para medir la calidad de video percibido por un usuario. Este algoritmo es desarrollado por Netflix y lo que busca es correlacionar los puntajes de MOS mediante aprendizaje de máquina y acercar un puntaje más real de la percepción humana y al igual que las demás métricas, es de referencia completa. VMAF fusiona métricas de calidad mediante "machine learning", para este proceso usa un modelo llamado Máquina de Soporte de Vectores (SVM) y así obtiene la calificación de calidad percibida por el usuario. Dentro del modelo SVN las métricas que están fusionadas se basan en algoritmos de fidelidad de imagen y de señal temporal; estos algoritmos son: SNR anti-ruido (AN-SNR), Medida de ítem detallada (DLM), Fidelidad de la información Visual (VIF), Diferencia media de píxeles cúbicos (MCPD) (Rassool, 2017). La Figura 4 muestra el proceso general con el cual se obtiene el puntaje QoE para VMAF.

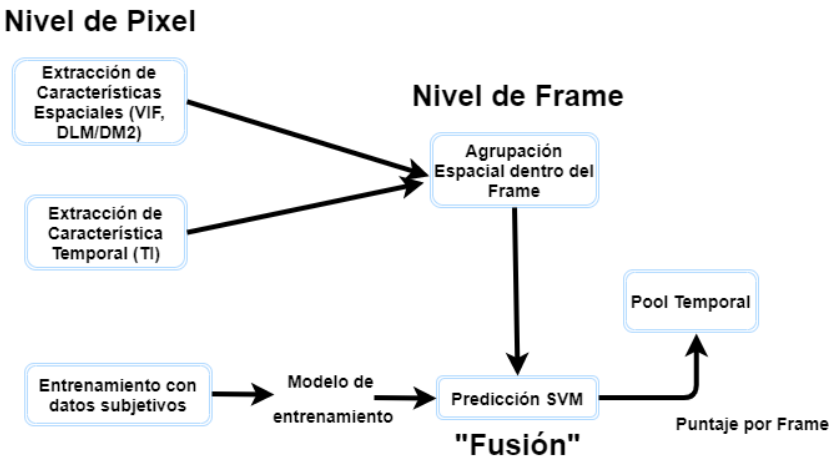

Figura. 4 - Arquitectura VMAF (Zhi Li et al., 2018)

El esquema presentado se puede dividir en dos etapas que posteriormente se fusionan. En la Figura 5 y Figura 6 se observa los procesos del esquema general de VMAF para obtener su puntaje final. La primera etapa (Figura 5), representa el proceso que se realiza sobre cada frame con métodos objetivos. Es decir, se evalúa contraste, frecuencia espacial, orientación, luminiscencia, entre otros. A esta etapa se la denomina Human Visual System Modeling (HVSM) (Zhi Li, et al., 2018).

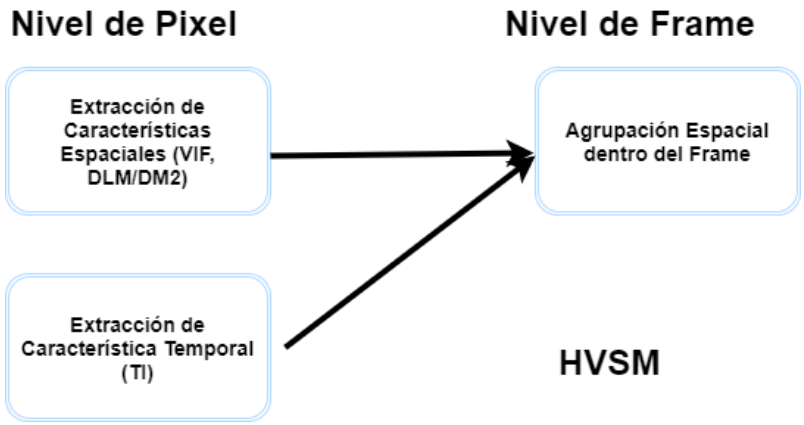

Figura. 5 – Primera etapa Human Visual System Modeling (HVSM) (Zhi Li, et al., 2018).

La segunda etapa que se observa en la Figura 6, es el proceso de denominado Machine Learning y se aplica el método de regresión para SVM. Para el entrenamiento de datos y obtener dicha puntuación se lo realiza con métodos subjetivos. La recolección de datos subjetivos se realiza con diferentes individuos de universidades en un ambiente adecuado, bajo estándares de distancia para la visualización del video, al igual que estándares de iluminación. Los sujetos de prueba visualizaron videos con una duración de 10 segundos, posteriormente votaron de acuerdo a la calidad percibida. Para el método de voto subjetivo escogieron a la Calificación de Categoría Absoluta (ACR). Este método está compuesto de 5 categorías, tales como: excelente, bueno, justo, regular y malo. La sesión de recolección de datos subjetivos tiene una duración total de 30 minutos (Zhi Li, et al., 2018).

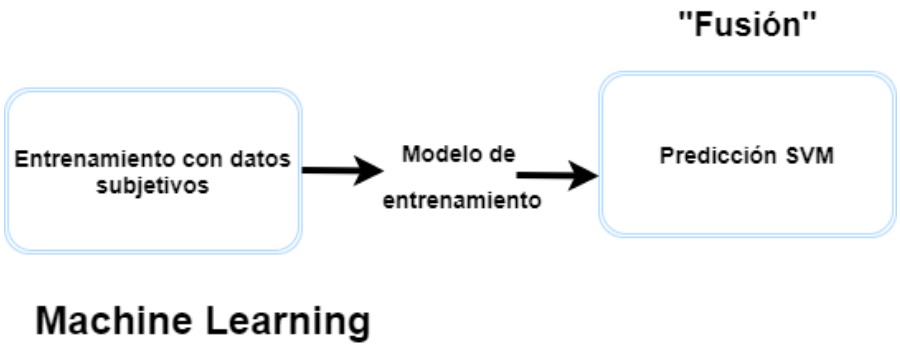

Figura. 6 - Segunda etapa Machine Learning (Zhi Li, et al., 2018).

En la Figura 7 se puede observar el mapeo de la escala ACR a la escala VMAF, en la que el puntaje final es sobre 100 y está la divide en 5 partes, al igual que ACR. Con crea el rango entre cada puntaje de ACR y VMAF (Zhi Li, et al., 2018).

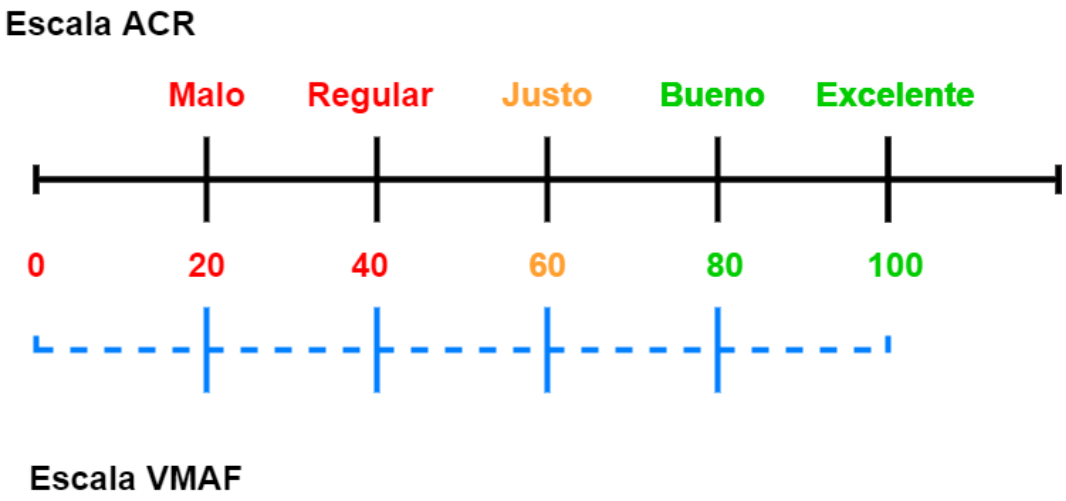

Figura. 7 - Mapeo del puntaje de escala ACR hacía Puntaje de escala VMAF (Zhi Li, et al., 2018).

3.3. Denegación de Servicios DoS

Un ataque DoS sucede cuando se realiza la interrupción del servicio desde una sola máquina (Yihunie et al., 2018). y se atacan objetivos, tales como la memoria del servidor, o espacio de disco, a la comunicación o infraestructura de red, a la capacidad de un nodo crítico como Servidor de Nombre de Dominio DNS o Sistema de Prevención de Intrusiones (IPS) (Yevsieieva & Helalat, 2017). Si un atacante quiere ser más agresivo y no ser detectados por firewalls o Sistemas de Detección de Intrusos (IDS) se usa métodos de Ataques Distribuidos de Denegación de servicio DDOS (Yevsieieva & Helalat, 2017). Los ataques con DDoS se inician desde varios dispositivos que han sido comprometidos de esta manera el nivel de saturación del ataque será mayor hacía el objetivo (Yihunie et al., 2018). En la Figura 8 se observa la arquitectura de un ataque DDoS (Douligeris & Mitrokotsa, 2004). La arquitectura está compuesta por: el atacante, quien es el nodo de inicio de un ataque DDoS. En el segundo nivel del árbol, los controladores son host que ya están comprometidos por algún tipo de ataque de DDoS y son capaces de controlar agentes. Posteriormente, los agentes son los responsables de generar el flujo de tráfico agresivo hacia la víctima. Por último la víctima es quién recibe el ataque (Douligeris & Mitrokotsa, 2004).

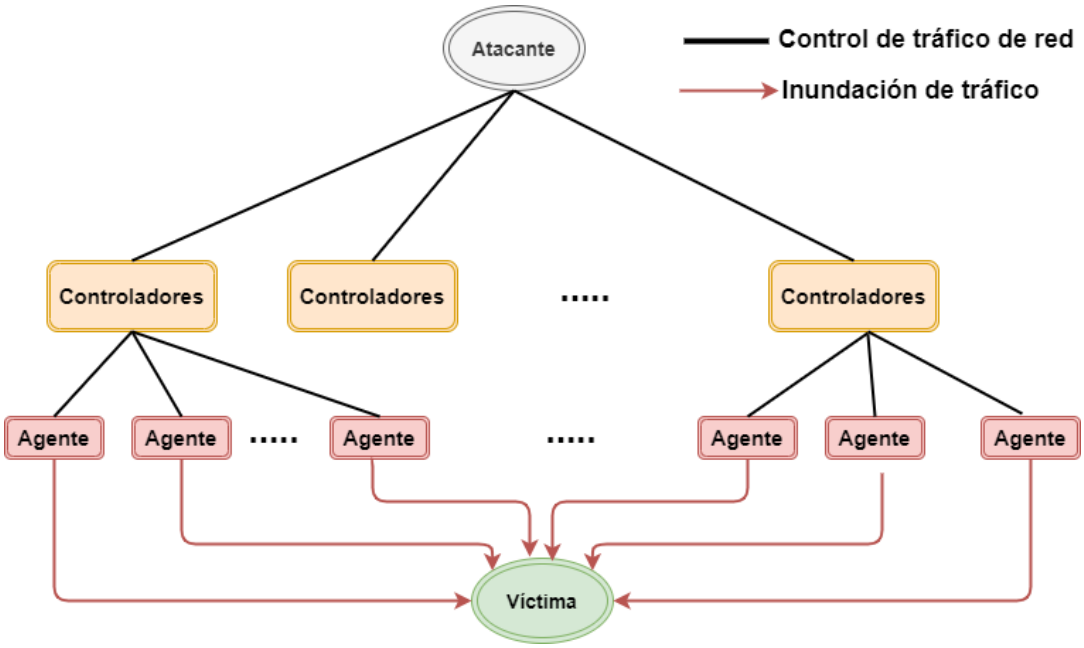

Figura. 8 - Arquitectura de un ataque DDoS (Douligeris & Mitrokotsa, 2004).

3.4. Tipos de ataques DoS

3.4.1. Ping de la Muerte (POD)

En este ataque el nodo de origen envía paquetes que contienen un mensaje volcado de forma masiva y continuamente hacía el host de la víctima. Este ataque se puede observar en la Figura 9 (Ramachandran & Shanmugam, 2017).

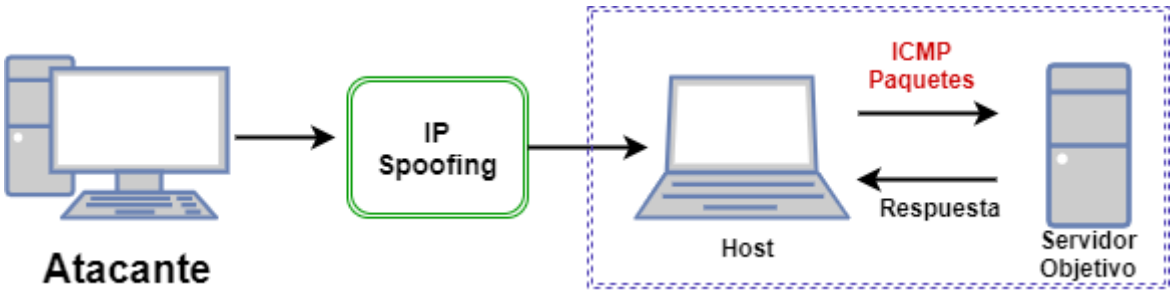

Figura. 9 - Ping de la Muerte (POD) (Ramachandran & Shanmugam, 2017).

3.4.2. HTTP Unbearable Load King (Hulk)

Como se observa en la Figura 10 el atacante envía una gran cantidad solicitudes HTTP hacía un host en particular. El contenido de los paquetes contiene la dirección de origen, destino, longitud del paquete, TTL, entre otros (Ramachandran & Shanmugam, 2017).

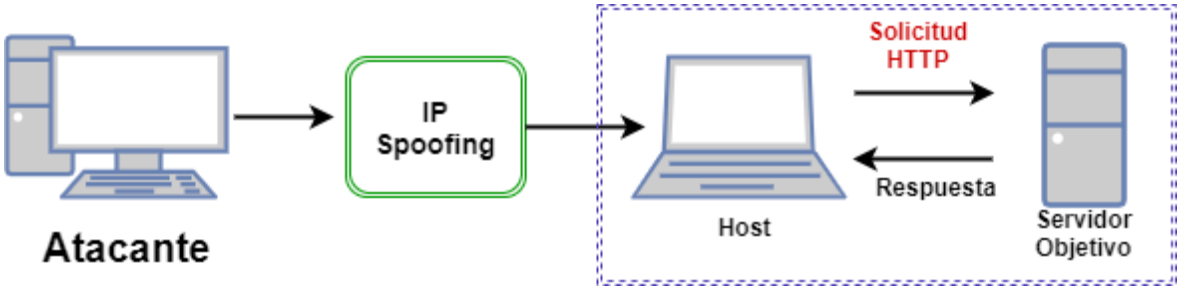

Figura. 10 - Http Unbearable Load King (Hulk) (Ramachandran & Shanmugam, 2017).

3.4.3. SYN Flood

Este ataque se basa en la implementación de hand-shake de tres vías entre los hosts. Como se muestra en la Figura 11 se realiza una solicitud para entablar una comunicación entre los hosts. Posteriormente, se envía un acuse de recibo y una solicitud a un host en particular. En este tipo de ataque se vuelve más agresivo cuando existe suplantación de direcciones IP (Ramachandran & Shanmugam, 2017).

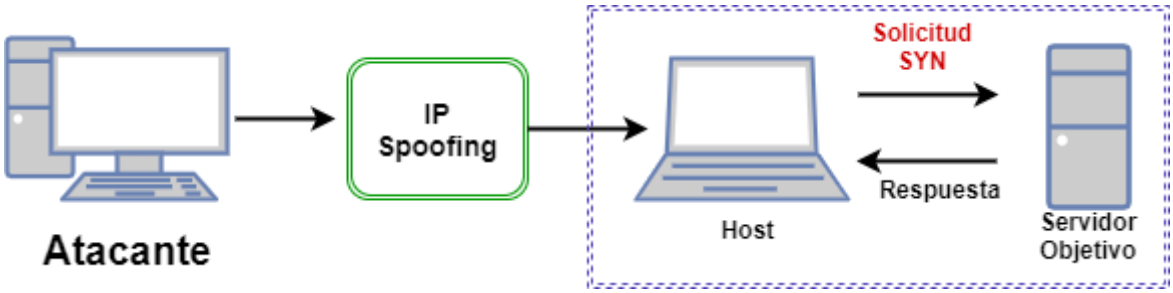

Figura. 11 - Syn Flood (Ramachandran & Shanmugam, 2017)

3.4.4. Ataque Smurf

Es uno de los ataques en donde una organización se ve inundada por numerosas cantidades de mensajes ping. Un ataque de Smurf satura el ancho de banda de la víctima, la misma que tiende a tardar en responder o en el peor de los casos no lo hace, tal como se aprecia en la Figura 12 (Ramachandran & Shanmugam, 2017).

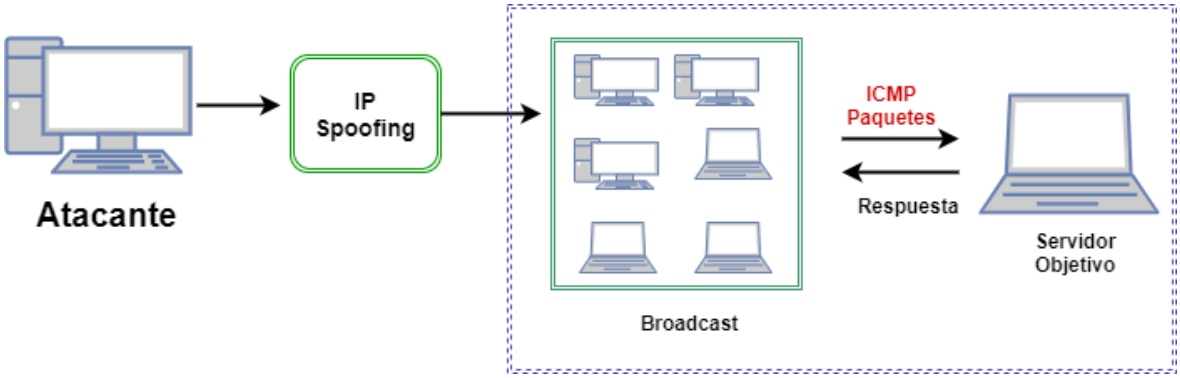

Figura. 12 - Ataque Smurf (Ramachandran & Shanmugam, 2017)

4. TRABAJOS RELACIONADOS

En el presente capítulo se revisan los últimos avances que integran los conceptos analizados anteriormente.

4.1. Servicios Streaming Multimedia y SDN

Algunos trabajos han implementado redes SDN para streaming multimedia a fin de mejorar la experiencia de usuario y control de flujo de datos, permitiendo una mejor experiencia en la calidad de transmisión. Por ejemplo, en (Yu et al., 2015) se propone el enrutamiento dinámico para transmisión de video con soporte de QoS, en una red SDN. Los resultados obtenidos de la propuesta indican que existe una reducción de 77,3% de paquetes perdidos. También en (Bentaleb et al., 2017) se propone la arquitectura SDNHAS como una mejora en streaming de video sobre HTTP (HAS). Esta arquitectura aprovecha los beneficios principales de SDN para maximizar la QoE en HAS, en donde cada capa de la arquitectura SDN está compuesta por un conjunto de módulos, componentes y clases que tienen una determinada función. Los resultados mostraron que la arquitectura planteada tiene grandes beneficios, aumentando la estabilidad de video en un 32%, la equidad de QoE en un 33% y el uso de recursos de red en un 29%, en comparación de otras técnicas de streaming multimedia (MDASH, QDASH, PANDA, FESTIVE). Así mismo, en (Zhao & Medhi, 2017) se describe como mejorar los flujos de transmisión a través de una red SDN para diferentes tecnologías, ya sea: 3D, realidad virtual, videos de 360° o contenidos en 2D; por lo que concluyeron que su trabajo se enfocará en plataformas de transmisión de realidad virtual en 360° usando DASH.

En (Karakus & Durresi, 2017b), propone una categorización de mecanismos para medir la QoS en redes SDN/Openflow, como se muestra en la Figura 13. De izquierda a derecha las dos primeras categorías están enfocadas en el funcionamiento del enrutamiento. Los dos siguiente se basan en la reserva de recursos, la programación de los paquetes para soporte de QoS. La quinta categoría aborda los aspectos de QoE del sistema, la sexta gira en torno en cuanto a al marco de monitoreo de red. Por último, mecanismos orientados a QoS se centra en gestión de políticas de calidad del servicio, las extensiones del banco de pruebas de calidad del servicio (Karakus & Durresi, 2017b).

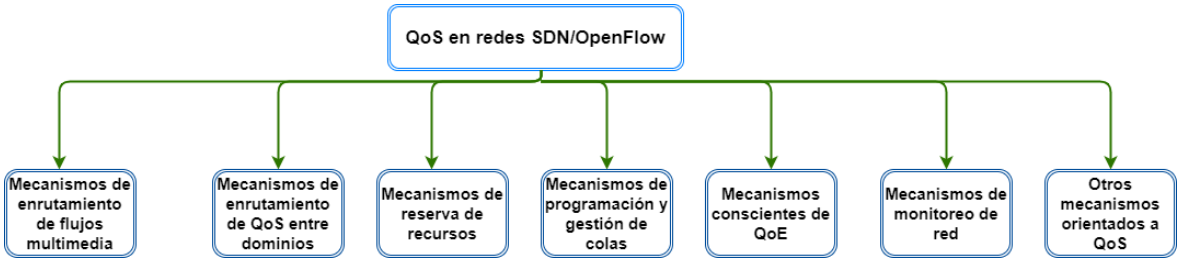

Figura. 13 - Clasificación de QoS en SDN/Openflow (Karakus & Durresi, 2017b)

4.2. Ataques DoS en SDN

Existen estudios relevantes sobre ataques de DoS en redes SDN. Por ejemplo, en (Polat & Polat, 2017) se estudia el efecto de un ataque de DoS que va dirigido hacia el controlador Opendaylight (ODL) y POX en una red SDN. Los resultados muestran que mientras el ataque se mantenía activo sobre dos hosts, el ancho de banda TCP entre estos dos nodos se degradaba de 1.2 Mbps con tendencia a 0 Mbps, en un tiempo de 5.5 segundos y el tiempo de respuesta de cualquiera de los hosts lo hacían luego de 120ms. Adicionalmente, en (Goksel & Demirci, 2019), se analiza cómo influye un ataque de inundación de paquetes en el controlador. En el estudio se logra distinguir a los atacantes de los usuarios normales mediante recuento de paquetes irregulares, para esto se determina un número tope de paquetes mediante un algoritmo. Para lograr determinar el número tope usan dos métodos de medición de equidad que es el de Jain y de entropía. Los autores concluyen que el método de Jain tiene un umbral del 10% mayor a la de entropía para detección de recuento de paquetes anómalos. Por su parte, flowsec (Kuerban et al., 2016) implementa medidas de control de envío de paquetes al controlador. Las medidas de control son estadísticas del conmutador que se generan mediante el algoritmo flowsec, de este modo se calcula el ancho de banda del controlador de forma dinámica. De igual manera, SDN-Guard Application (Dridi & Zhani, 2016) intenta mitigar los efectos de un ataque mejorando el rendimiento del controlador. La mitigación se realiza redirigiendo el tráfico malicioso dinámicamente con ajustes de tiempo de espera de flujo, y agregando reglas asociadas al flujo de tráfico malicioso, en consecuencia, se logra reducir el número de entradas al controlador.

Según la información que se dispone, hasta el momento no se ha analizado a fondo los efectos de un ataque DoS sobre la calidad de servicio en aplicaciones de streaming multimedia en redes SDN

5. ARQUITECTURA PROPUESTA

En el presente proyecto se propone una arquitectura enfocada en facilitar el análisis de ataques DoS sobre una red SDN. Nuestra arquitectura se encuentra detallada en la Figura 14, la misma que permitirá tener una Red SDN completamente operativa, enviar tráfico de red entre sus dispositivos y permitir la ejecución de una aplicación de streaming. Así mismo, la arquitectura presenta módulos que permiten efectuar un ataque de DoS mientras el sistema está funcionando, recolectar los resultados de la ejecución, y sintetizar la información por medio de gráficos. Los componentes principales son:

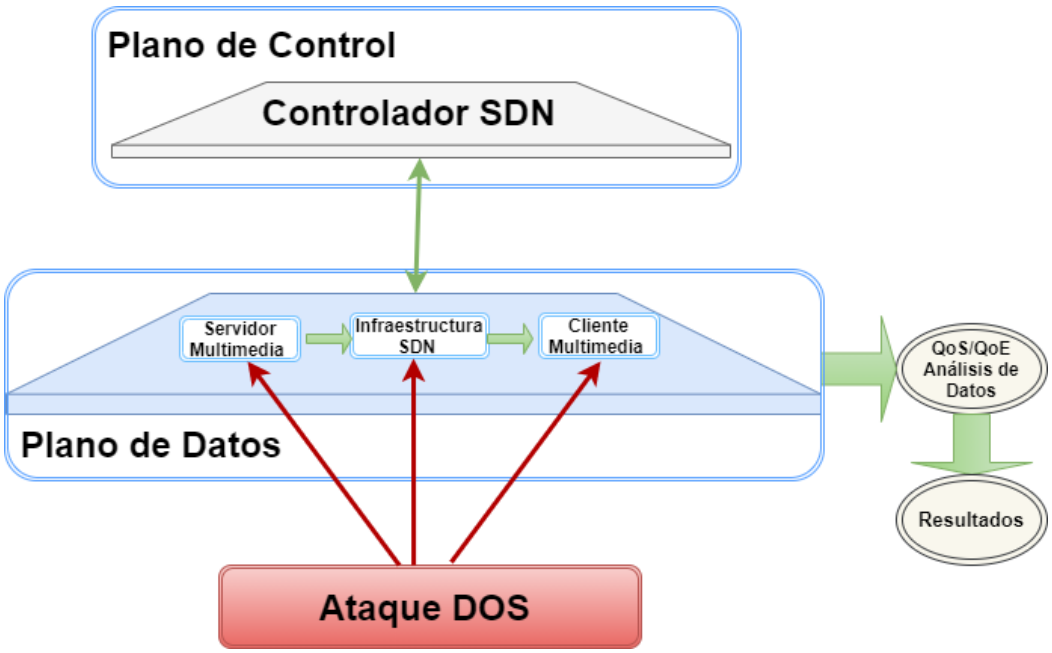

Figura. 14 - Arquitectura Propuesta

- **Infraestructura SDN. -** La infraestructura SDN contiene a los dispositivos de hardware de la red (plano de datos) interconectadas en una topología específica. Dichos dispositivos son los encargados de transmitir los paquetes además de comunicarse con el controlador SDN para recibir instrucciones del plano de control.
- **Controlador SDN. -** Es el encargado de administrar el flujo de datos dentro de la red SDN. Además, el controlador tiene la información de la topología y controla la configuración de cada uno de los swichtes que forman parte de la infraestructura. El controlador recibe información sobre el origen y destino de los paquetes entrantes, ejecuta algoritmos para buscar la ruta óptima y envía órdenes a los switches para enviar los paquetes a su destino.

- **Servidor Multimedia. -** El servidor multimedia es el encargado de enviar el video hacía el cliente vía streaming. Para la entrega de video y audio se usa un protocolo a nivel de aplicación, como por ejemplo el protocolo Real-Time Protocol (RTP). RTP tiene un diseño "end-to-end" para la transmisión de los medios en tiempo real y se ejecuta sobre el Protocolo de Datagramas de Usuario (UDP) dentro de la red SDN.
- **Cliente Multimedia. -** Es el receptor del video enviado por el servidor multimedia por un determinado puerto y dirección IP previamente configurado. La recepción del video será continuamente almacenada para su posterior análisis. El cliente también trabaja con los protocolos utilizados por el servidor (RTP y UDP).
- **Ataque DoS. -** Es un host conectada a la red SDN el cual realiza diferentes ataques a los componentes de la red: servidor multimedia, cliente multimedia y los switches SDN. Cabe mencionar que la red SDN necesita de la integración entre plano de datos y control, por lo que el ataque a cualquiera de estos elementos también influyen sobre el rendimiento del controlador SDN.

6. IMPLEMENTACIÓN

En la Figura 15 se detallan las herramientas que se utilizan para la implementación en cada capa de la arquitectura. Los detalles técnicos de las herramientas de software utilizadas se describen en la Tabla 2. Entre los datos más importantes se incluyen: la versión del software, al que módulo pertenece de la arquitectura y características generales del software.

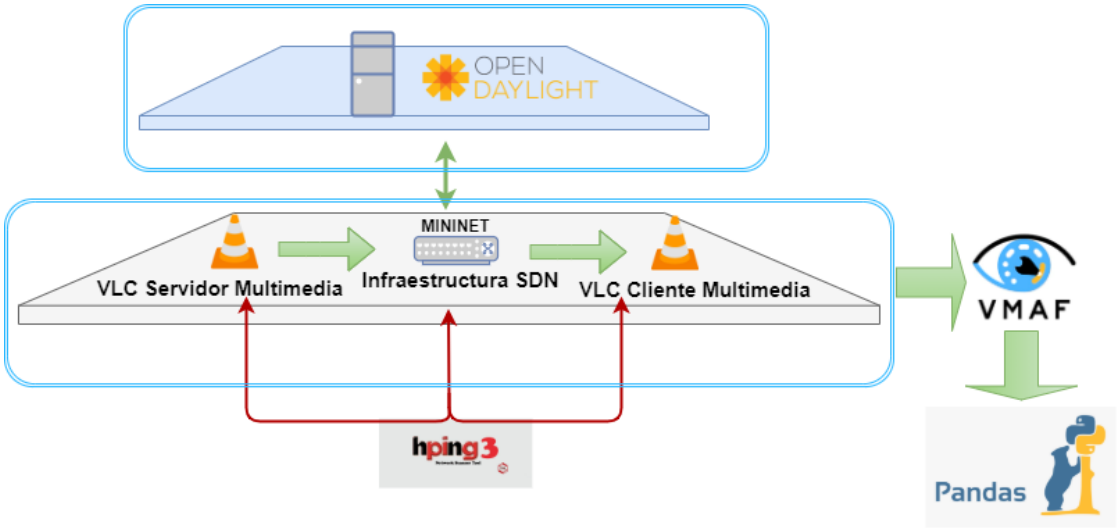

Figura. 15 - Implementación de la Arquitectura

Tabla 2 - Herramientas utilizadas en la implementación.

MÓDULO	HERRAMIENTA	VERSIÓN	DETALLES
Controlador SDN	OpenDaylight	0.13.1	• Open source • Controlador de Openflow • Posee Interfaz Gráfica (Polat & Polat, 2017).
Multimedia Server, Cliente	VLC	3.0.8	• Open source • MP4, AVI, MP3 formatos de contenedor multimedia. • RTP, UDP, HTTP protocolos de streaming (M. Li et al., 2018).

Ataque DoS	HPING 3	3	• Open source • Pruebas de firewall • Analizador de paquetes de TCP/IP (Polat & Polat, 2017).
Infraestructura SDN	Mininet	2.3.0d6	• Open source • Emula redes SDN • Switches soportan protocolos OpenFlow (Lantz & O'connor, 2015).
QoS/QoE: Análisis de datos	VMAF	vmafv4	• Open source • Métricas QoS: PSNR, SSIM, ADM2 • Usa técnicas de machine learning. • Su métrica VMAF tiene como objetivo la QoE .(Rassool, 2017)
Resultados	Pandas/Python	3.7	• Open source • Lenguaje interpretado, no compilado. • Es multiplataforma (Wes McKinney, Joris Van den Bossche, Tom Augspurger, n.d.).

En el proyecto se usa Mininet para emular el despliegue de redes basado en SDN y Openflow. Mininet permitirá crear una topología de red SDN de forma interactiva en hardware virtual. Cabe resaltar que Mininet tiene la capacidad de crear contenedores de red mediante una API de Python. Además, evita la necesidad de instalación, configuración y administración de múltiples sistemas de orquestación (Bob Lantz, 2015). Los hosts de Mininet son grupos de procesos conectados a interfaces virtuales de Ethernet y usan las características de los name-spaces de Linux (Kaur et al., 2014).

El código y las instrucciones de implementación de la presente arquitectura se encuentran disponibles en (Mantilla, 2021). Adicionalmente el manual se encuentra en el Anexo 1, donde están las indicaciones para la instalación del ambiente y ejecución de los scripts.

7. TOPOLOGÍA Y EXPERIMENTOS

La eficacia y usabilidad de la arquitectura se demuestra mediante experimentos y tráfico real de una aplicación de streaming. La topología utilizada corresponde a un Datacenter, tal como se indica en la Figura 16, es así que, de arriba hacia abajo tiene 3 secciones: Switch Core, unión de comunicación entre S2 y S5 y sus capas inferiores. La segunda capa Switch de Distribución acepta el tráfico de la capa inferior, es decir de la capa de acceso y facilita la entrada de dispositivos de red. Cada switch de la capa de acceso posee dos hosts. Para los experimentos el host H1 es el servidor multimedia, H7 es el cliente multimedia que recibe el video streaming y H4 es el host que realiza el ataque de DoS. De esta manera el envío del video recorrerá más enlaces de la topología y habrá más interacción con el controlador, como consecuencia se obtiene resultados más fiables.

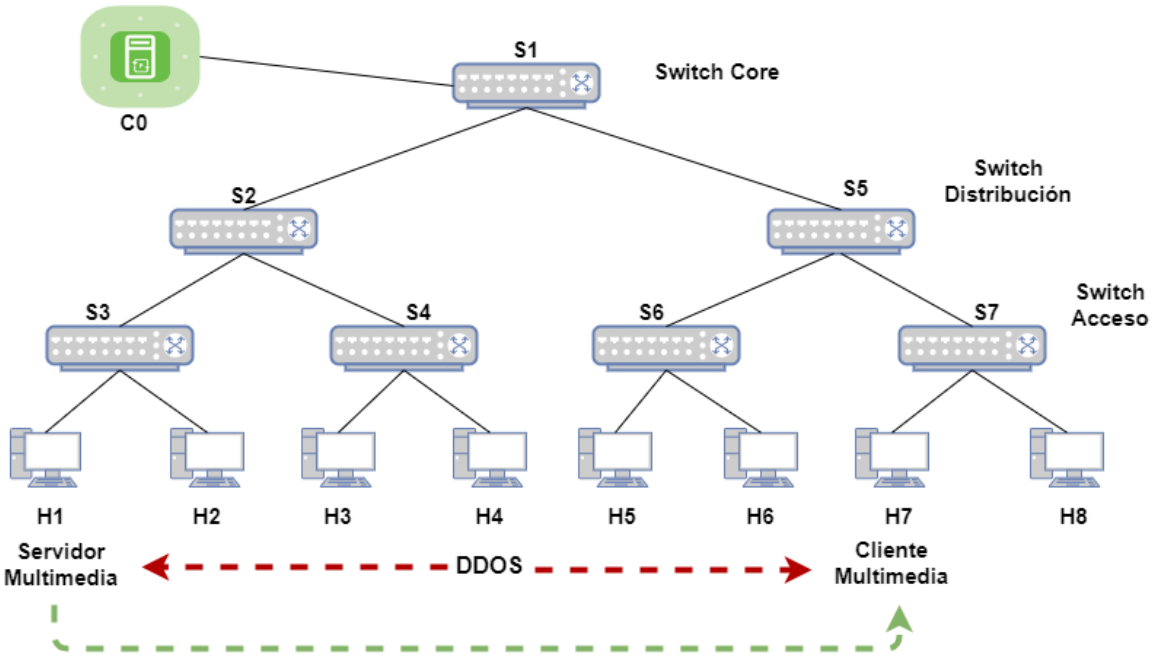

Figura. 16 - Topología de Experimentación - Data Center

Cabe mencionar que, el ancho de banda de cada enlace trabaja a 1MB/s, este valor se escogió en base a la tasa máxima de transmisión de bits del video de prueba que es 717 Kbps. La transmisión de bits es obtenida de la metadata en el análisis con la herramienta VMAF. Para las pruebas se utilizó el video BuckBunny.mp4 que está codificado en MPEG4 (Test-Videos, n.d.,2010). El video fue descargado de un repositorio para pruebas de QoS, tiene un tamaño es de 2.1MB, una resolución de 320x240, una duración total de 26 segundos y 401 frames.

8. EXPERIMENTACIÓN

Para los experimentos se tiene dos escenarios. En el primer escenario o escenario ideal, no existe ninguna alteración en la red o en los hosts, y el video se transmite de manera normal. En el segundo escenario se realizan dos tipos de ataques de DoS: spoofing y flooding. Las diferencias entre estos tipos de ataques se aprecian en la Figura 17.

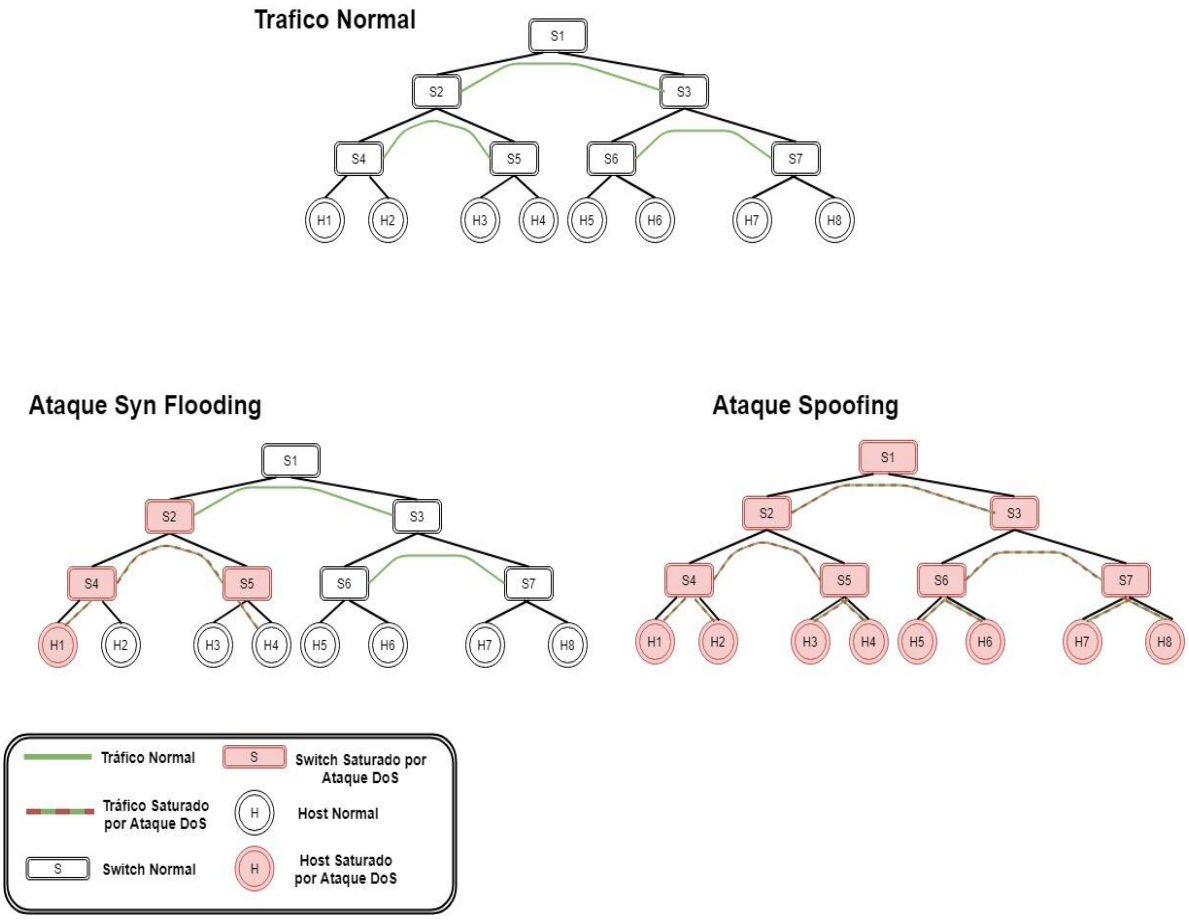

Figura. 17 – Diferencia entre un ataque DoS Syn Flooding y Spoofing

Los dos ataques empiezan a los 8 segundos de iniciar el video y tiene como objetivo a 3 actores de la arquitectura (Servido Multimedia, Cliente Multimedia e infraestructura de la red -switches SDN-). En la Tabla. 3 se detallan las características de cada uno de los ataques. Estos ataques han sido seleccionados debido a que son los que normalmente causan un mayor daño en una red. En el caso de redes SDN, debido a que el plano de datos tiene que enviar las cabeceras de paquetes nuevos hacia el plano de control, es muy probable que el controlador SDN se sature rápidamente causando un incremento en el

tiempo de respuesta y ocasionando una degradación del servicio en la red (Polat & Polat, 2017).

Tabla 3 - Tipos de ataques del segundo escenario.

ATAQUE	FUENTE	OBJETIVO	DETALLES
Flooding	H4	S4, S2, S3, H1	• Ataque SYN/TCP • Tamaño de ventana 64 • Tamaño del cuerpo del paquete 500000 bytes.
Flooding	H4	S4, S2, S1, S5, S6, S7, H7	• Ataque SYN/TCP • Tamaño de ventana 64. • Tamaño del cuerpo del paquete 500000 bytes.
Spoofing	H4	S1, S4, S2, S3, S5, S6, S7, Controlador	• Ataque SYN/TCP • Tamaño de ventana 64 • Tamaño del cuerpo del paquete 500000 bytes. • Se asignará una IP falsa a la fuente. • Envío de paquetes randómico a diferentes hosts

9. RESULTADOS Y DISCUSIÓN

Una vez enviada información multimedia por la red SDN en condiciones normales y bajo ataque DoS, los resultados son procesados y analizados con la herramienta VMAF. Debido a que se ejecutan varias tareas en la misma máquina virtual (mininet, vlc, vmaf), se pueden producir variaciones en los resultados. Con el objetivo de reducir las distorsiones se utiliza el método montecarlo, el cual consiste se realizan varias repeticiones (10) para cada escenario y se obtiene el promedio de las variables QoS y QoE calculadas.

9.1. Métrica ADM2

La Figura 18 muestra el promedio de ADM2, el cual mide la pérdida de detalle que afecta la visibilidad del contenido. El puntaje del parámetro ADM2 se encuentra entre los valores de 0 a 1, donde 1 significa que no ha perdido detalle de imagen y 0 es el puntaje más bajo para la pérdida de detalle imagen; este puntaje es por cada frame del video. De igual manera en la Figura 18, para cada ataque se tiene una representación lineal y color diferente. Cuando se realiza un streaming normal (línea verde) su tendencia hasta el frame 360 tiene un puntaje de 1, posteriormente en el frame 361 baja a un puntaje de 0,99445. Para el segundo caso de ataque al servidor multimedia (línea roja), hasta el frame 384 tiene un de puntaje 1 y luego empieza a degradarse hasta un factor 0,999914 en el frame 385.
El ataque al cliente (línea negra), antes de los 8 primeros segundos el puntaje es el alto y en el frame 169 cae con un puntaje de 0,98455. Por último, el ataque de spoofing (línea azul), posterior a los 8 segundos, en el frame 167 empieza la degradación de calidad y su puntaje es de 0,997332.

Los resultados demuestran que la métrica ADM2 es afectada gravemente a los 8 segundos de empezar el ataque DoS. Los ataques más efectivos para el parámetro ADM2 son: el que va dirigido al cliente (línea negra - frame 169) y spoofing (línea azul - frame 167). Sin embargo, el ataque al cliente tiene mayor repercusión, pero a partir del frame 350 tienden a mantenerse sobre un valor de 0,5 y 0,6 hasta finalizar el video.

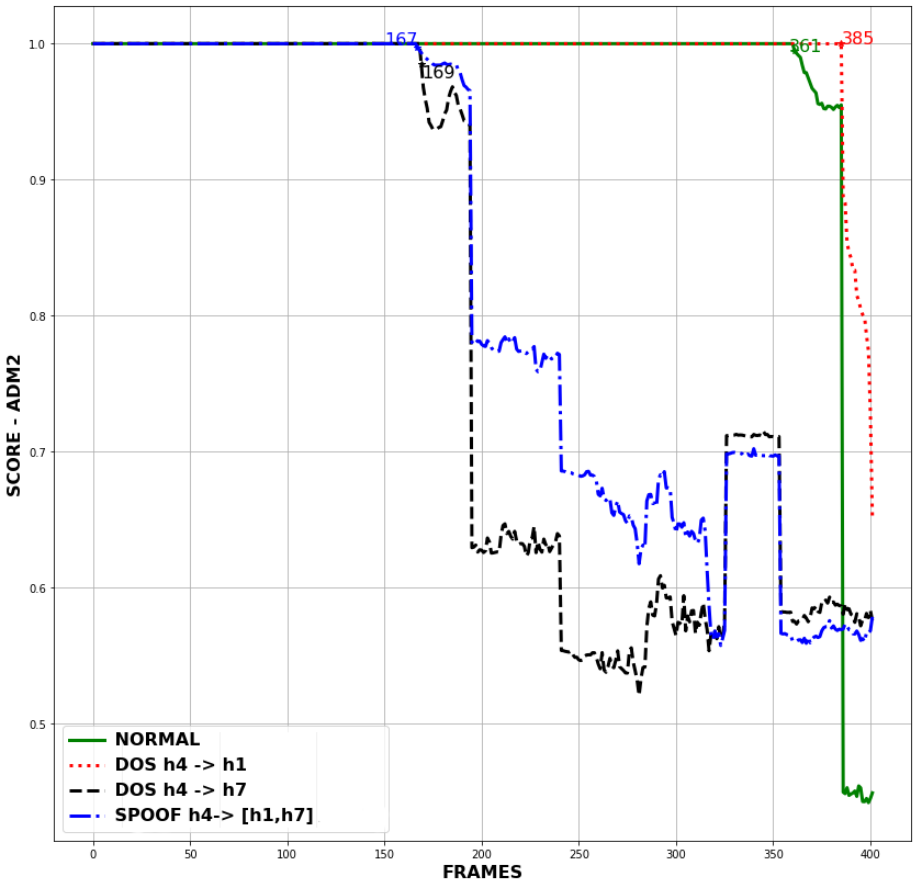

Figura. 18 - Efecto de los ataques DoS en términos DLM/ADM2

9.2. Métrica PSNR

En la Figura 19 la métrica PSNR evalúa la relación señal de pico a ruido, que afecta la fidelidad de la representación del frame. Su puntaje está definido en el rango de valores de 0 a 60: en el cual, 60 representa su valor máximo que significa que su calidad para este parámetro es alta, mientras tanto que 0 es la más baja con una calidad mala. En condiciones normales (línea verde), hasta el frame 357 tiene un puntaje de valor máximo, para el frame 358 tiende a degradarse y tiene 52,311966. En el ataque al servidor ataque al servidor (línea roja), inicia con un el puntaje más alto hasta el frame 384 y a partir del frame 385 tiende a caer con un puntaje de 59,721358.

Además, en el ataque al cliente, a partir del frame 169 tiende a decaer con un puntaje de 39,0031. Por último, en el ataque spoofing (línea azul) se observa que los primeros 8 segundos tiene un puntaje de 60 y posterior a esto el frame 167 tiene un puntaje de 57,769966.

Al comprar los resultados de PSNR con ADM2 se observa que en el paso del frame 168 al 169 se tiene un descenso precipitado. Así mismo el ataque de spoofing tiene una tendencia de caída menos drástica, que pasa de un valor de 60 a un valor 57,77 que representa 3.71% de pérdida de calidad. Por ello concluimos que un ataque al cliente en un streaming es más efectivo que un ataque de spoofing.

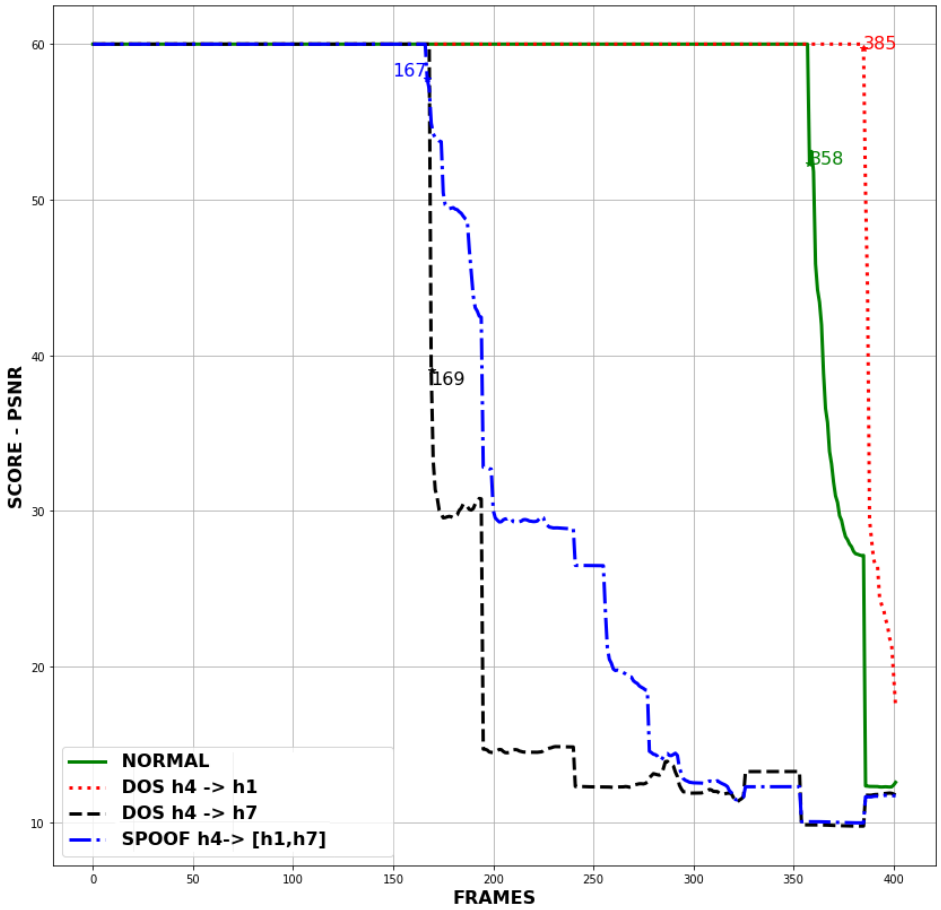

Figura. 19 - Efecto de los ataques DoS en términos de PSNR

9.3. Métrica SSIM

SSIM evalúa la distorsión de dos imágenes en base a los valores de luminancia, contraste y textura. En la Figura 20 se observa que, al igual que ADM2 se evalúa entre los rangos de 0 a 1, siendo 1 el puntaje más alto para esta métrica y 0 el más bajo. Los puntajes obtenidos de los frames en el momento exacto que empieza la degradación de calidad en cada ataque para SSIM son: streaming normal (línea verde), frame 358 su puntaje es 0,999504. Ataque al servidor (línea roja), frame 385 tiene un puntaje de 0,99997. Mientras que cuando se

realiza un ataque al cliente (línea negra) se obtiene un puntaje de 0,99016 en el frame 169. Por último, en un ataque de spoofing (línea azul) en el frame 167 se consigue un valor de 0,994091. En la Figura 19, se observa que a partir del frame 167 (línea azul)- frame 169 (línea - negra) se ven afectados por el ataque DoS. Posteriormente en el rango de frames, entre 300 y 350 toma un leve repunte durante unos segundos, pero en los dos casos de ataque (DoS Cliente y spoofing) su puntaje SSIM vuelve a caer. Además, entre los frames 326 y 352, al igual que ADM2 tienen una leve alza en los puntajes que oscilan entre 0,61742 y 0,61616.

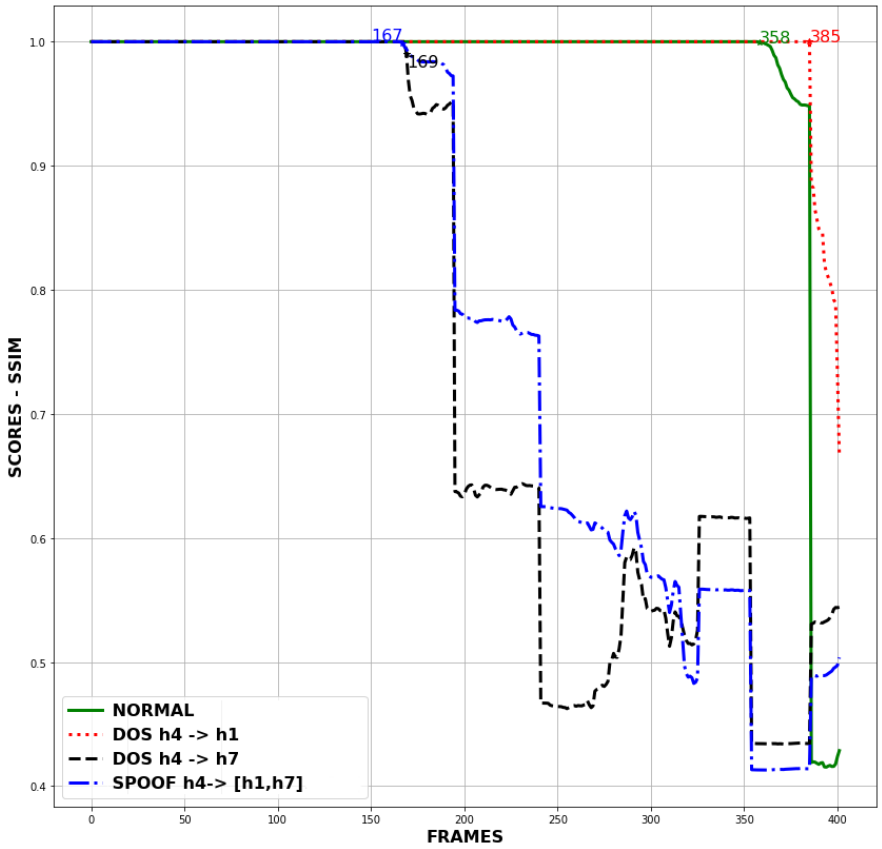

Figura. 20 - Efecto de los ataques DoS en términos de SSIM

9.4. Métrica VMAF

Ya que la resolución del video es de 320x240, se recomienda evaluar a VMAF como: el promedio de las muestras del puntaje final del video completo (Zhi Li, Anne Aaron, Ioannis Katsavounidis, n.d., 2016). La Tabla 4 presenta los valores promedio y las desviaciones estándar de cada una de las pruebas realizadas. De acuerdo a la Tabla. 4, presenta los valores promedio y las desviaciones estándar de cada una de las pruebas realizadas. En la misma se puede inferir que, los puntajes más bajos se obtienen con el ataque de spoofing (54.781) y al cliente (46.56). En consecuencia, la calidad de video es "Regular". Cabe resaltar que, en las Figura17, Figura18 y Figura19, el ataque al servidor (línea roja), se mantiene sobre las puntuaciones altas, que el streaming normal (línea verde) en los últimos frames. Además, se tiene un puntaje mayor en cada una de las métricas ya sean de QoS (ADM2, PSNR, SSIM) ó QoE (VMAF), como se muestra en la Tabla 4. Este comportamiento se debe a que la entrega de paquetes en el escenario uno tiende a llegar a H7 en menos tiempo y, para el segundo escenario (línea roja), el host H7 se queda esperando la recepción de paquetes completos. Por este motivo se concluye que existe un mayor retardo en la línea roja (DoS h4->h1) que en la línea verde (Normal) y por lo tanto el puntaje de la métrica en el ataque al servidor multimedia no se ve afectado.

Tabla 4 - Promedio y Desviación Estándar de métricas

MÉTRICA / **ATAQUE**	**ADM2**		**PSNR**		**SSIM**		**VMAF**	
	Avg.	**D.E**	**Avg.**	**D.E**	**Avg.**	**D.E**	**Avg.**	**D.E**
Normal	0,97	0,1	56,32	0,062	0,97	0,000041	93,62	0,063
DoS h4->h1	0,99	0,03	58,68	0,31	0,99	0,008	96,73	0,83
DoS h4->h7	0,79	0,19	33,65	9,30E-16	0,76	8,90E-17	46,56	4,70E-15
DoS Spoof h4->[h1,...h7]	0,82	0,17	37,49	5,46	0,78	0,06	54,781	11,33

10. CONCLUSIONES Y RECOMENDACIONES

En el presente proyecto se propone una arquitectura que permite analizar la degradación de la calidad de streaming multimedia cuando existe un ataque DoS en una red SDN. La arquitectura se encuentra compuesta por 5 módulos y su implementación se realizó utilizando herramientas OpenSource.

Mediante las pruebas efectuadas se obtuvieron los resultados esperados en una topología tipo datacenter que demostraron que un ataque de DoS al cliente multimedia y un ataque de spoofing a la red SDN son los que más afectan la calidad percibida por el usuario en términos de QoS/QoE (VMF, PSNR, SSIM, ADM2); siendo así que el parámetro más afectado para un ataque al cliente es PSNR con una pérdida del 56,1% en QoS, mientras que para QoE con VMAF 53,44%. De la misma manera en spoofing PSNR obtiene una degradación de 34,52% en QoS y en QoE 45.22% (VMAF).

Cabe resaltar que los parámetros ADM2, SSIM, PSNR y VMAF, pueden variar por diferentes factores como: asignación de recursos a la máquina virtual donde se efectuará el experimento, velocidad de ancho de banda de enlaces en la red SDN y configuración de la herramienta para realizar el ataque DoS. Además, el comportamiento de entrega de paquetes en la red SDN también dependerá de la configuración del controlador, ya que es quién administra el flujo de datos en SDN.

El software se encuentra compartido en código abierto y puede ser utilizado por la comunidad científica para para ampliar el número de ataques, así como implementar nuevas topologías de red. Los resultados servirán para el diseño de nuevos algoritmos que permitan la detección y mitigación temprana de ataques.

REFERENCIAS BIBLIOGRÁFICAS

Ab, T., & Asmi, S. El. (2017). *Objective and subjective measurement QoE in SDN networks*. 1401–1406.

Bentaleb, A., Begen, A. C., Zimmermann, R., & Harous, S. (2017). SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming. *IEEE Transactions on Multimedia*, *19*(10), 2136–2151. https://doi.org/10.1109/TMM.2017.2733344

Brunnström, K., Beker, S. A., De, K., Dooms, A., Egger, S., Garcia, M.-N., Hossfeld, T., Jumisko-Pyykkö, S., Keimel, C., & Larabi, M.-C. (2013). Qualinet White Paper on Definitions of Quality of Experience Qualinet White Paper on Definitions of Quality of Experience Output from the fifth Qualinet meeting, Novi Sad. *European Network on Quality of Experience in in Multimedia Systems and Services (COST Action IC 1003)*, *March*, 26.

Caraguay, Á. L. V., Ludeña-González, P. J., Tandazo, R. V. T., & López, L. I. B. (2018). SDN/NFV architecture for IoT networks. *WEBIST 2018 - Proceedings of the 14th International Conference on Web Information Systems and Technologies*, *Webist*, 425–429. https://doi.org/10.5220/0007234804250429

Cherif, W. (2013). *Adaptation of the context based on the Quality of Experience in the Internet networks of the future*. UNIVERSIT Y OFRENNES.

Danda B. Rawat, & Reddy, S. R. (2017). *Software Defined Networking Architecture, Security and Energy Efficiency: A Survey*. *19*, 22.

Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Computer Networks*, *44*(5), 643–666. https://doi.org/10.1016/j.comnet.2003.10.003

Dridi, L., & Zhani, M. F. (2016). SDN-Guard: DoS Attacks Mitigation in SDN Networks. *Proceedings - 2016 5th IEEE International Conference on Cloud Networking, CloudNet 2016*, 212–217. https://doi.org/10.1109/CloudNet.2016.9

Egilmez, H. E., & Tekalp, A. M. (2014). Distributed QoS architectures for multimedia streaming over software defined networks. *IEEE Transactions on Multimedia*, *16*(6), 1597–1609. https://doi.org/10.1109/TMM.2014.2325791

Gabriel, F., Nguyen, G. T., Schmoll, R. S., Cabrera, J. A., Muehleisen, M., & Fitzek, F. H.

P. (2018). Practical deployment of network coding for real-time applications in 5G networks. *CCNC 2018 - 2018 15th IEEE Annual Consumer Communications and Networking Conference*, *2018-Janua*, 1–2. https://doi.org/10.1109/CCNC.2018.8319320

Goksel, N., & Demirci, M. (2019). DoS attack detection using packet statistics in SDN. *2019 International Symposium on Networks, Computers and Communications, ISNCC 2019*, 1–6. https://doi.org/10.1109/ISNCC.2019.8909114

Guofei Gu. (2008). *CORRELATION-BASED BOTNET DETECTION IN ENTERPRISE NETWORKS*. Georgia Institute of Technology.

Hyun, D., Kim, J., Hong, D., & Jeong, J. (2017). SDN-based network security functions for effective DDoS attack mitigation. *International Conference on Information and Communication Technology Convergence: ICT Convergence Technologies Leading the Fourth Industrial Revolution, ICTC 2017*, *2017-Decem*, 834–839. https://doi.org/10.1109/ICTC.2017.8190794

Karakus, M., & Durresi, A. (2017a). A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN). *Computer Networks*, *112*, 279–293. https://doi.org/10.1016/j.comnet.2016.11.017

Karakus, M., & Durresi, A. (2017b). Quality of Service (QoS) in Software Defined Networking (SDN): A survey. *Journal of Network and Computer Applications*, *80*, 200–218. https://doi.org/10.1016/j.jnca.2016.12.019

Kaur, K., Singh, J., & Ghumman, N. S. (2014). Mininet as Software Defined Networking Testing Platform. *International Conference on Communication, Computing & Systems (ICCCS–2014), August 2014*, 3–6.

Kuerban, M., Tian, Y., Yang, Q., Jia, Y., Huebert, B., & Poss, D. (2016). FlowSec: DOS attack mitigation strategy on SDN controller. *2016 IEEE International Conference on Networking Architecture and Storage, NAS 2016 - Proceedings*, 7–8. https://doi.org/10.1109/NAS.2016.7549402

Kunwar Pal, Mahesh Chandra Govil, M. A. and, & Chawla, T. (n.d.). *A Survey on Adaptive Multimedia Streaming*.

Lantz, B., & O'connor, B. (2015). A mininet-based virtual testbed for distributed SDN development. *SIGCOMM 2015 - Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 365–366. https://doi.org/10.1145/2785956.2790030

Lee, C., Woo, S., Baek, S., Han, J., Chae, J., & Rim, J. (2018). Comparison of objective quality models for adaptive bit-streaming services. *2017 8th International Conference on Information, Intelligence, Systems and Applications, IISA 2017*, *2018-Janua*, 1–4. https://doi.org/10.1109/IISA.2017.8316385

Li, J., Krasula, L., Le Callet, P., Li, Z., & Baveye, Y. (2018). Quantifying the Influence of Devices on Quality of Experience for Video Streaming. *2018 Picture Coding Symposium, PCS 2018 - Proceedings*, 308–312. https://doi.org/10.1109/PCS.2018.8456304

Li, M., Yeh, C. L., & Lu, S. Y. (2018). Real-time QoE monitoring system for video streaming services with adaptive media playout. *International Journal of Digital Multimedia Broadcasting*, *2018*. https://doi.org/10.1155/2018/2619438

Liberato, A., Martinello, M., Gomes, R. L., Beldachi, A. F., Salas, E., Villaca, R., Ribeiro, M. R. N., Kondepu, K., Kanellos, G., Nejabati, R., Member, S., Gorodnik, A., & Simeonidou, D. (2018). *RDNA : Residue-Defined Networking Architecture*. *15*(4), 1473–1487.

Liotou, E., Samdanis, K., Pateromichelakis, E., Passas, N., & Merakos, L. (2018). QoE-SD N APP: A Rate-guided QoE-aware SDN-APP for HTTP Adaptive Video Streaming. *IEEE Journal on Selected Areas in Communications*, *36*(3), 598–615. https://doi.org/10.1109/JSAC.2018.2815421

Mantilla, C. (2021). *Arquitectura de medición del impacto de ata-ques dos en la qos de servicios multimedia en redes sdn*. https://github.com/laboratorioAI/DoS_IoT_SDN_Framework.git

Oktian, Y. E., Lee, S., & Lee, H. (2014). Mitigating Denial of Service (DoS) attacks in OpenFlow networks. *International Conference on ICT Convergence*, 325–330. https://doi.org/10.1109/ICTC.2014.6983147

Polat, H., & Polat, O. (2017). The effects of DoS attacks on ODL and POX SDN controllers. *ICIT 2017 - 8th International Conference on Information Technology, Proceedings*, 554–558. https://doi.org/10.1109/ICITECH.2017.8080058

Ramachandran, S., & Shanmugam, V. (2017). Impact of DoS Attack in Software Defined Network for Virtual Network. *Wireless Personal Communications*, *94*(4), 2189–2202. https://doi.org/10.1007/s11277-016-3370-1

Rassool, R. (2017). VMAF reproducibility: Validating a perceptual practical video quality metric. *IEEE International Symposium on Broadband Multimedia Systems and*

Broadcasting, BMSB. https://doi.org/10.1109/BMSB.2017.7986143

Sachdeva, M., Singh, G., Kumar, K., & Singh, K. (2010). Measuring Impact of DDOS Attacks on Web Services. *International Journal of Computer Science and Information Security*, *5*(9), 392–400.

Schaller, S., & Hood, D. (2017). Software defined networking architecture standardization. *Computer Standards and Interfaces*, *54*, 197–202. https://doi.org/10.1016/j.csi.2017.01.005

Stefano Petrangeli, Tim Wauters, Rafael Huysegems, Tom Bostoen, F. D. T. (2016). *Software-defined network-based prioritization to avoid video freezes in HTTP adaptive streaming*. https://doi.org/10.1002/nem.1931

Tamayo y Tamayo, M. (1980). *Metodología formal de la investigación científica*. http://www.worldcat.org/profiles/afgomez/lists/2904204

Test-Videos. (n.d.). *Buck Bunny Video*. https://test-videos.co.uk/bigbuckbunny/mp4-h264

Vranješ, M., Rimac-Drlje, S., & Grgić, K. (2013). Review of objective video quality metrics and performance comparison using different databases. *Signal Processing: Image Communication*, *28*(1), 1–19. https://doi.org/10.1016/j.image.2012.10.003

Wes McKinney, Joris Van den Bossche, Tom Augspurger, S. H. (n.d.). *Pandas*. pandas.pydata.org

Yevsieieva, O., & Helalat, S. M. (2017). Analysis of the impact of the slow HTTP DOS and DDOS attacks on the cloud environment. *2017 4th International Scientific-Practical Conference Problems of Infocommunications Science and Technology, PIC S and T 2017 - Proceedings*, *2018-Janua*, 519–523. https://doi.org/10.1109/INFOCOMMST.2017.8246453

Yihunie, F., Abdelfattah, E., & Odeh, A. (2018). Analysis of ping of death DoS and DDoS attacks. *2018 IEEE Long Island Systems, Applications and Technology Conference, LISAT 2018*, 1–4. https://doi.org/10.1109/LISAT.2018.8378010

Yu, T. F., Wang, K., & Hsu, Y. H. (2015). Adaptive routing for video streaming with QoS support over SDN networks. *International Conference on Information Networking*, *2015-Janua*, 318–323. https://doi.org/10.1109/ICOIN.2015.7057904

Zhao, S., & Medhi, D. (2017). SDN-Assisted adaptive streaming framework for tile-based immersive content using MPEG-DASH. *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks, NFV-SDN 2017*, *2017-Janua*, 1–6. https://doi.org/10.1109/NFV-SDN.2017.8169831

Zhi Li, Anne Aaron, Ioannis Katsavounidis, A. M. and M. M. (n.d.). *Toward A Practical Perceptual Video Quality Metric*. https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

Zhi Li, Christos Bampis, Julie Novak, Anne Aaron, Kyle Swanson, A. M. and J. D. C. (2018). *VMAF: The Journey Continues*. https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12

ANEXOS

ANEXO I MANUAL DE INSTALACIÓN Y USO

1. MANUAL DE INSTALACIÓN

Este manual de instalación permitirá instalar el software necesario para realizar las pruebas del proyecto.

1.1. Instalación de Git

Los pasos de instalación de git son los siguientes

- Instalar git con el siguiente comando: ***sudo apt-get install git*.**
- En la Figura 1 se observa el comando para saber la versión instalada.

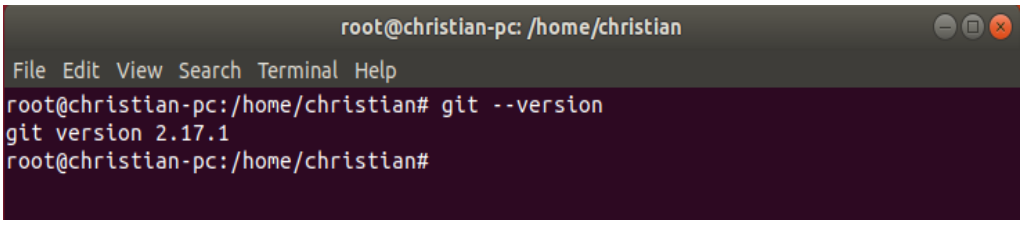

Figura. 21 - Comprobación de instalación de git

1.2. Instalación de Mininet

Para la instalación de mininet seguir las siguientes instrucciones:

- Descargar del repositorio git con el siguiente comando:
  git clone git://github.com/mininet/mininet
- Ingresar a la carpeta cd mininet.
- Instalar mininet con ***mininet/util/install.sh***
- Se realiza un test con el siguiente comando para hacer una prueba ***"sudo mn -switch ovsbr --test pingall"***, como se muestra en la Figura 2

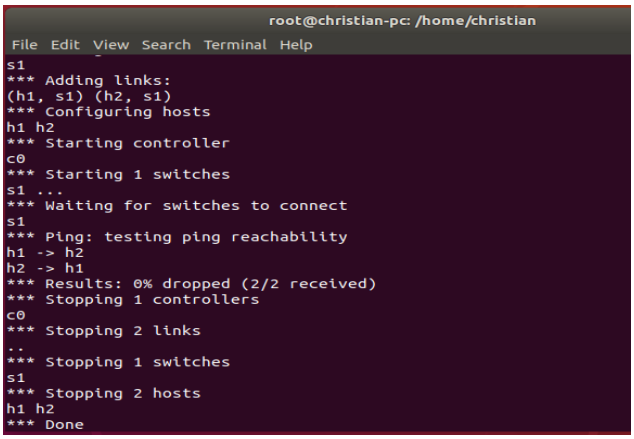

Figura. 22 - Prueba de instalación Mininet

1.3. Instalación de FFMPEG

Para la instalación del códec FFMPEG se realiza los siguiente:

- Instalar el códec ffmpeg con el siguiente comando:
  sudo apt-get update -qq && sudo apt-get -y install autoconf automake build-essential cmake git-core libass-dev libfreetype6-dev libgnutls2 dev libsdl2-dev libtool libva-dev libvdpau-dev libvorbis-dev libxcb1-dev libxcb-shm0-dev libxcb-xfixes0-dev meson ninja-build pkg-config texinfo wget yasm zlib1g-dev
- Instalar las dependencias de ffmpeg:
  mkdir -p ~ / ffmpeg_sources ~ / bin
- En la Figura 3 se observa como comprobar que se encuentre instalado ffmpeg y su versión.

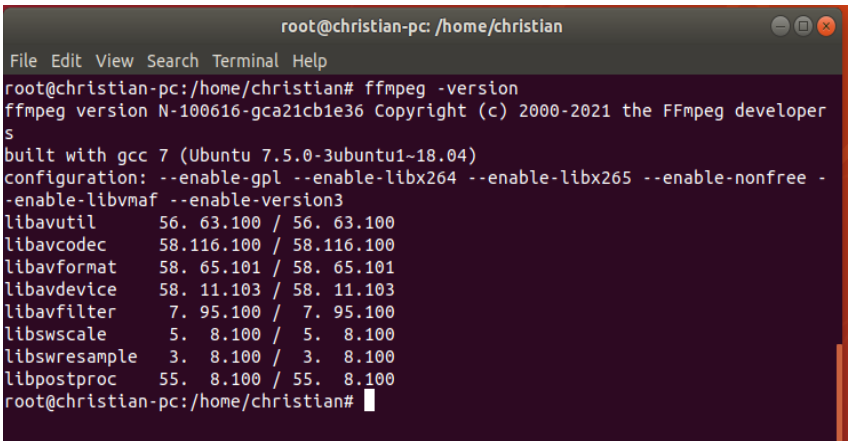

Figura. 23 – FFMPEG

1.4. Instalación de VMAF

La herramienta para el cálculo de puntaje de QoS/QoE se realiza de la siguiente manera:

- Descargar vmaf de los repositorios de github:
  wget https://github.com/Netflix/vmaf/archive/v1.5.2.tar.gz
- Descomprimir el archive tar.gz:
  untar -xvf https://github.com/Netflix/vmaf/archive/v1.5.2.tar.gz
- Instalar dependencias:
  a. ***sudo apt update***
  b. ***sudo apt install python3 python3-pip python3-setuptools python3-wheel ninja-build doxygen***
  c. ***pip3 install meson***
  d. ***pip3 install Cython***
  e. ***pip3 install numpy***

- Instalar VMAF:
    a. ***cd vmaf-1.5.2/***
    b. ***sudo make***
    c. ***sudo make install***

- La variable de entorno apuntará a la librería libvmaf con el siguiente comando:

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib/x86_64-linux-gnu/

1.5. Instalación de VLC

La instalación del reproductor de video para streaming, se lo hace con el siguiente paso:

- Instalar vlc con el siguiente comando:

sudo apt-get install vlc

1.6. Descarga de recursos del repositorio

En el siguiente link se puede descargar del repositorio el proyecto https://github.com/laboratorioAI/DoS_IoT_SDN_Framework, donde se encuentra el video de referencia y el archivo commands.txt el cual iniciara el servicio de vlc; también contiene dos carpetas, la carpeta Scripts Streaming attack DoS tiene los scripts de python donde está configurado la red SDN, el ataque de DoS y la configuración para realizar el streaming entre dos nodos de la red SDN. La segunda carpeta Bash contiene los archivos bash quienes serán los encargados de realizar el proceso de ejecución del ambiente de pruebas. En la Figura 4 se observa la distribución del repositorio.

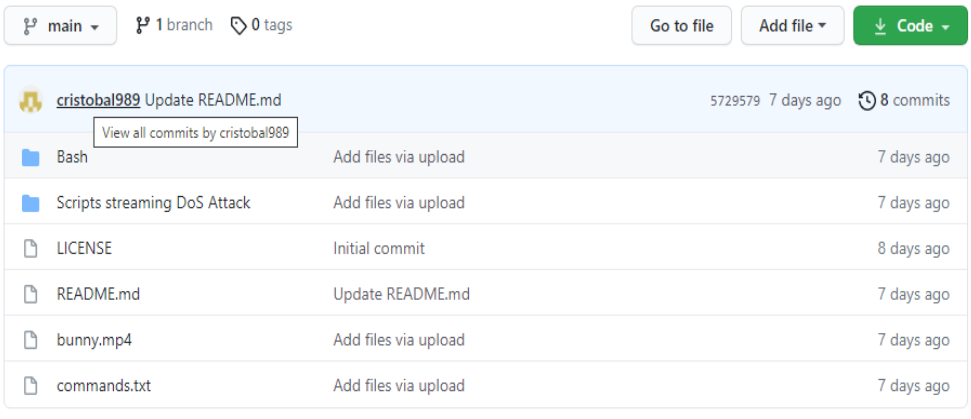

Figura. 24 - Repositorio github del proyecto.

2. MANUAL DE USO

Este manual permitirá hacer uso de los scripts y bash para las pruebas concernientes del proyecto.

a) En la Figura 5 se observa el esquema general del archivo bash para la ejecución de pruebas. En la programación de este archivo se tiene una variable “contador”, quién es el que determinará el número de veces que realizará el streaming y el análisis de los vídeos. Es importante determinar la dirección de donde se tomará el video de referencia inicial y el video luego del streaming.
También hay que determinar la dirección donde se guardará los datos y serán de tipo json. Así mismo la dirección de los scripts .py deben ser escritos en el bash.

Open — tesisBashforNorm.sh — /data/Descargas/media_org/scripts — Save

```
#!/bin/bash

contador=2
for i in $(seq 1 $contador)
do
        sudo mn -c
        sleep 1s
        mkdir /data/Descargas/media_org/temp
        chmod 777  /data/Descargas/media_org/temp
        echo "START NETWORK MININET"
        sudo python auto_vlc_StreambandwidthNorm.py
        sudo cp /data/Descargas/media_org/temp/mininet_video_h7_1234.mp4 /data/Descargas/media_org/media/
stream/mininet_video_h7_1234_$i.mp4
        sleep 2s
        sudo ffmpeg -r 24 -i /data/Descargas/media_org/media/bunny.mp4 -r 24 -i /data/Descargas/
media_org/media/stream/mininet_video_h7_1234_$i.mp4 -lavfi  "[0:v]setpts=PTS-STARTPTS[reference];
[1:v]scale=320:240:flags=bicubic,setpts=PTS-STARTPTS[distorted]; [distorted]
[reference]libvmaf=psnr=1:ssim=1:log_fmt=json:model_path=/usr/local/share/model/other_models/
nflxall_vmafv4.pkl:log_path=/data/Descargas/media_org/dataset/datos/logpr$i.json" -f null -
        sleep 15s
        sudo rm -rf /data/Descargas/media_org/temp/

done
```

Figura. 25 – Esquema bash para pruebas

b) Se ejecuta el comando de la Figura 6 en consola para que se inicien las pruebas y presionamos la tecla enter.

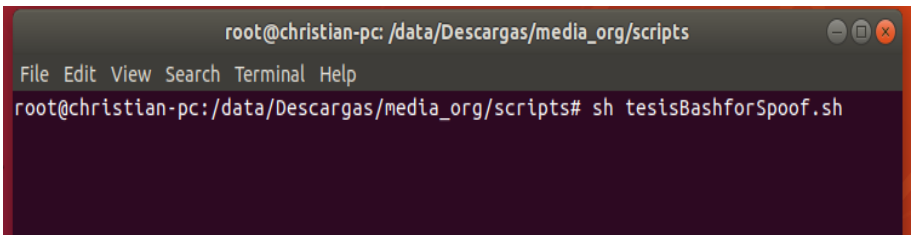

Figura. 26 - Comando para iniciar pruebas

c) Como se puede observar en la Figura 7, automáticamente las pruebas empezaran y se ejecutaran los scripts con extensión py donde se encuentra configurado: la topología tipo Datacenter en una red SDN, video streaming con vlc y la configuración de un ataque DoS a los 8 segundos.

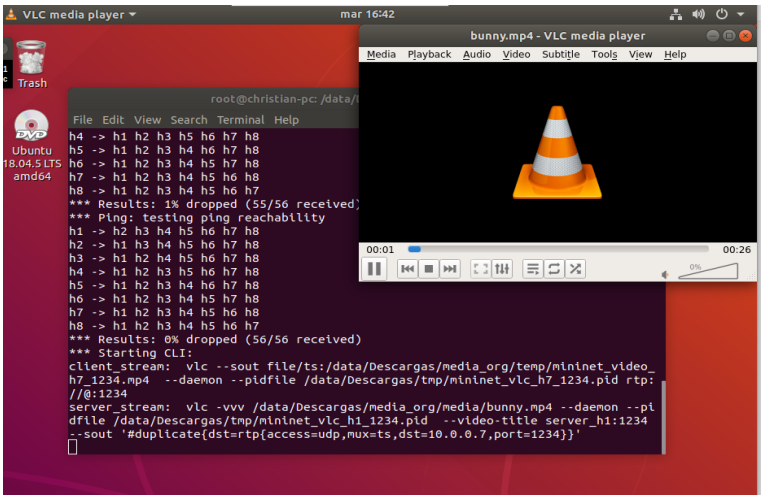

Figura. 27 -Ejecución de pruebas automáticas

d) En la Figura 8 observamos que determina el puntaje VMAF con este archivo bash.

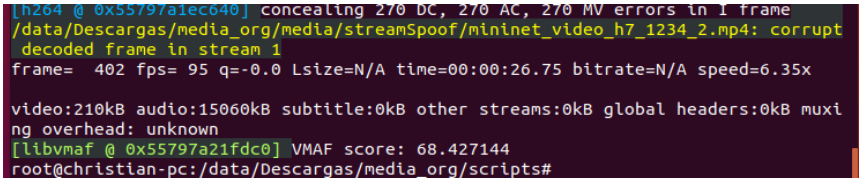

Figura. 28 - Puntaje VMAF

e) En la Figura 9 se observa el archivo de análisis del video por frame


```
{
  "frameNum":2,
  "metrics":{
    "adm2":1.0,
    "motion2":0.74968,
    "psnr":60.0,
    "ssim":0.99998,
    "vif_scale0":1.0,
    "vif_scale1":1.0,
    "vif_scale2":1.0,
    "vif_scale3":1.0,
    "vmaf":98.46889
  }
},
{
  "frameNum":3,
  "metrics":{
    "adm2":1.0,
    "motion2":1.34085,
    "psnr":60.0,
    "ssim":0.99998,
    "vif_scale0":1.0,
    "vif_scale1":1.0,
    "vif_scale2":1.0,
    "vif_scale3":1.0,
    "vmaf":99.28386999999999
  }
},
{
```


Figura. 29 - Ejemplo de análisis de video por frame

f) Para graficar los resultados se debe transformar el archivo de resultados de tipo json a una extensión de tipo csv. En la variable path colocamos la ubicación del archivo, como se observa en la Figura 10. Es importante colocar los enunciados del archivo csv en la función plt.plot(), está función está definida de la siguiente manera plt.plot(*eje x, eje y, color, nombre de la etiqueta*). A continuación, se muestra un ejemplo del uso de la función, por ejemplo:
"plt.plot(df['frames__frameNum'],df['metrica_psnrFinalNorm'],color='green', label = 'PSNR NORMAL', linewidth =3)"

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Mon Jan 18 14:45:34 2021

@author: christian
"""

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
path = "/data/Descargas/media_org/dataset/Graficos/FinalTest/TestFinalmetrics/GraphFinalPSNR.csv"
df = pd.read_csv(path)
fig = plt.figure(figsize=(14,14))
ax = plt.subplot(1, 1, 1)
#plt.scatter(df['frames__frameNum'],df['metrica_vmafFinalNorm'],color='green', label = 'SSIM NORMAL')
plt.plot(df['frames__frameNum'],df['metrica_psnrFinalNorm'],color='green', label = 'PSNR NORMAL',
linewidth =3 )
#plt.axvline(361, color='green')
plt.plot(358,52.311966,color='green', marker='*')
ax.annotate('358',xy=(358,52.311966), color = 'green', fontsize= 16)

plt.plot(df['frames__frameNum'],df['metrica_psnrFinalDoS_h4_h1'], color='red', label='PSNR DOS h4 -> h1',
linestyle=':', linewidth =3)
#plt.scatter(df['frames__frameNum'],df['metrica_vmafFinalDoS_h4_h1'], color='red')
#plt.axvline(385, color='red')
plt.plot(385,59.721358,color='red', marker='*')
ax.annotate('385',xy=(385,59.721358), color = "red", fontsize= 16)
```

Figura. 30 - Script de gráfica de resultados

g) Con la descripción de las variables y el uso de las funciones se ejecuta el programa y los resultados se observa en la Figura 11.

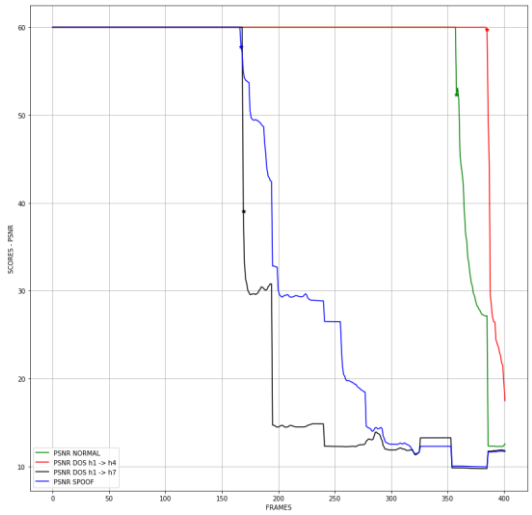

Figura. 31 - Ejemplo de script de gráfica de resultados