# Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería Electrónica y Telecomunicaciones

**Mejoramiento de la calidad de servicios (QoS) en aplicaciones de telediagnóstico basado en redes definidas por software (SDN): caso de estudio Clínica Monte Sinaí**

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones

**Autor:**

Verónica Aleksei Viloria Ramírez

**Director:**

Carlos Villie Morocho Zurita

ORCID: 0000-0002-8196-2644

**Codirector:**

Santiago Renán González Martínez

ORCID: 0000-0001-6604-889X

**Cuenca, Ecuador**

2023-09-16

**Resumen**

En un entorno en constante evolución, dónde la transmisión eficiente de datos médicos a distancia cobra relevancia, esta investigación se enfoca en evaluar métricas de calidad de servicios (QoS) mediante redes definidas por software (SDN). El objetivo principal es evaluar estas métricas con miras a mejorar la transmisión de paquetes en aplicaciones de telediagnóstico, un ámbito crítico para la QoS. Para ello, se establecieron objetivos específicos que van desde la identificación de métricas óptimas de QoS hasta el análisis comparativo de los resultados. De manera sistemática y rigurosa, se identificaron métricas óptimas de QoS para telediagnóstico y se midieron en la red de la Clínica Monte Sinaí (escenario real). Seguidamente, se realizó el diseño y la simulación de una red SDN empleando MiniNet y el controlador OpenDaylight (escenario simulado) y haciendo uso de la topología de la clínica, realizando experimentos en ambos escenarios con archivos de tamaño típico de imágenes médicas. Los resultados indican que el escenario SDN mejora la latencia al usar el protocolo TCP, logrando transmitir la imagen completa en la simulación, a diferencia del escenario real de la clínica. Además, al emplear UDP se observa una mejora en términos de ancho de banda, *jitter* y latencia en ciertos experimentos. En síntesis, esta investigación ofrece una comprensión profunda de cómo la implementación de SDN puede beneficiar la QoS en aplicaciones de telediagnóstico, al optimizar la transmisión de datos médicos a distancia.

*Palabras clave:* Redes Definidas por Software, telediagnóstico, telemedicina, MiniNet, OpenDaylight

**Abstract**

In a constantly evolving environment, where the efficient transmission of medical data over distance is gaining relevance, this research focuses on evaluating, quality of service (QoS) metrics using software-defined networking (SDN). The main objective is to evaluate these metrics with a view to improving packet transmission in telediagnostic applications, a critical area for QoS. To this end, specific objectives were established, ranging from the identification of optimal QoS metrics to the comparative analysis of the results. Systematically and rigorously, optimal QoS metrics for telediagnosis were identified and measured in the Monte Sinaí Clinic network (real scenario). Next, and SDN network was designed and simulated using MiniNet and the OpenDaylight controller (simulated scenario) and making use of the clinic's topology, performing experiments in both scenarios with typical file sizes of medical images. The results indicate that the SDN scenario improves latency when using the TCP protocol, managing to transmit the complete image in the simulation, unlike the real clinic scenario. In addition, when using UDP, an improvement in terms of bandwidth, jitter and latency is observed in certain experiments. In summary, this research provides an in-depth understanding of how SDN implementation can benefit QoS in telediagnostic applications by optimizing remote medical data transmission.

*Keywords:* Software Define Networking, quality of service, telediagnosis, telemedicine, MiniNet, OpenDaylight

**Índice de contenido**

**Índice de figuras**

**Índice de tablas**

**Agradecimientos**

Quiero agradecer a Dios por darme la oportunidad de poder lograr cumplir mi sueño de ser ingeniera después de migrar de mi país hace más de 6 años, luego de tener que dejar mi primera carrera de ingeniería a un 7mo ciclo por los problemas económicos, políticos y de seguridad que atraviesa mi país.

También quiero agradecer a mi familia por el apoyo, la confianza y paciencia que me dieron durante todo este tiempo.

Gracias a Pablo Andrés por creer en mí en los días dónde quería rendirme.

Y como dijo *Snoop Dogg*, “por último y no menos importante quiero agradecerme a mi misma... Quiero agradecerme a mi misma por confiar en mi, quiero agradecerme a mi misma por hacer todo este trabajo duro, quiero agradecerme a mi misma por no tener días libre, quiero agradecerme a mi misma por nunca renunciar, quiero agradecerme a mi misma por ser yo todo el tiempo.”

**Verónica Aleksei Viloria Ramírez**

Yo, Verónica Akesei Viloria Ramírez en calidad de autora y titular de los derechos morales, patrimoniales e intelectuales del trabajo de titulación “MEJORAMIENTO DE LA CALIDAD DE SERVICIOS (QOS) EN SERVICIOS DE TELEDIAGNÓSTICO BASADO EN REDES DEFINIDAS POR SOFTWARE (SDN): CASO DE ESTUDIO CLÍNICA MONTE SINAÍ”, manifiesto que la idea presentada en este trabajo de investigación fue planteada por la empresa INNTRATEC S.A.S. y mi persona. A petición de la empresa y la Corporación Médica Monte Sinaí esta idea está sujeta a cláusulas de confidencialidad, por lo que se mantendrá como confidencial, salvo que las otras partes, proporcionen el permiso expreso y por escrito para revelarla.

## Abreviaciones y Acrónimos

AMA - Asociación Médica Américana

API - Application Programming Interface

DICOM - Digital Imaging and Communications in Medicine

IA - Inteligencia Artificial

IP - Internet Protocol

ISO - International Organization for Standardization

IS4H - Information Systems for Health

MAC - Media Access Control

ODL - OpenDaylight

OF - OpenFlow

OF-S - OpenFlow Switch

OMS - Organización Mundial de la Salud

ONF - Open Networking Foundation

OPS - Organización Panamericana de Salud

PACS - Picture Archiving and Communication System

RT - Redes Tradicionales

SDN - Software Define Network

SLA - Service Level Agreement

TAC - Tomografía Axial Computarizada

TCP - Transmission Control Protocol

TI - Tecnología de la Información

TIC - Tecnología de la Información y Comunicación

UDP - User Datagram Protocol

UIT - Unión Internacional de Telecomunicaciones

VTN - Virtual Tenant Network

QoS - Quality of Service

# Introducción

## 1.1. Identificación del problema

En este capítulo se presenta una introducción a la temática del estudio, en particular se contextualiza en cómo funcionan hoy en día las redes, las diferentes problemáticas y limitaciones que éstas tienen, y se resaltan los conceptos de *Redes Definidas por Software* (SDN), *Calidad de Servicio* (QoS) y *Niveles de Servicios* (SLA).

Una casa de salud está formada por diferentes departamentos y requiere que su red se mantenga estable y en funcionamiento la mayor parte del tiempo posible. Además, dentro del área médica también se puede dividir en diferentes servicios, y éste necesita que su infraestructura de comunicación no se vea comprometida por posibles congestiones en el tráfico, y asegurar una priorización dentro de los diferentes servicios. Por lo general la casa de salud puede brindar, consultas externas, laboratorio, hospitalización, imagenología y farmacia. Sin embargo, la tendencia señala que uno de los próximos servicios que éstos pueden añadir es el de telemedicina, el cual es un enfoque de atención médica que utiliza las *Tecnologías de la Información y Comunicación* (TIC) para proporcionar atención médica a distancia. Dicho servicio permite la interacción entre médicos y pacientes a través de videoconferencias, aplicaciones móviles, plataformas en línea y otros medios digitales, facilitando la consulta médica, el diagnóstico, el monitoreo de pacientes y el intercambio de información médica, sin necesidad de una presencia física en la misma ubicación. Para poner en práctica éste enfoque es necesario contar con una red lo suficientemente robusta y controlada, entendiendo por robusta, la capacidad de resistir y recuperarse de fallos, interrupciones o condiciones adversas sin experimentar una degradación significativa en su rendimiento.

En éste trabajo se denominan a las Redes Tradicionales (RT), como infraestructuras de comunicación basadas en dispositivos físicos interconectados, como *routers* y *switches*, que utilizan protocolos estándar bien definidos basados en IP (*Internet Protocol*) que deben ser configurados previamente tales como, protocolos de enrutamiento, conmutación y control de red (Al-Haddad et al., 2021).

Una de las características principales de las RT es su principio complejo y estático lo que se resume en dificultades en la gestión, para las diferentes necesidades que puedan surgir, lo que implica, que éstas no brindan la flexibilidad de los entornos de red dinámicos que si permiten una gran cantidad de opciones para la configuración. Además, en los últimos años, las redes se han visto sometidas a un aumento en el tráfico de datos (Mohammadi y Javidan, 2018).

Para cualquier organización o empresa es importante establecer políticas de red, las cuales se refieren a las directrices y reglas establecidas para definir las condiciones de uso, acceso y seguridad de su infraestructura de comunicación. Estas políticas son un conjunto de normas y procedimientos que rigen la administración, configuración, operación, uso de los recursos, y buscan establecer un marco de referencia para el comportamiento y actividades relacionadas. Los administradores de red, usualmente deben configurar de manera manual e individual cada uno de los dispositivos de los distintos proveedores utilizando herramientas esenciales limitadas como por ejemplo, comandos de bajo nivel, lo que puede provocar una alta tasa de errores, además, del tiempo que se necesita invertir. Sin embargo, esto no asegura que los dispositivos tengan la flexibilidad suficiente para trabajar en entornos complejos con diferentes servicios y grandes flujos de datos (Al-Haddad et al., 2021), tampoco que se reconfigure de manera dinámica de acuerdo a las necesidades de los servicios con una diferencia considerable de tiempo entre las dos soluciones.

En estudios como (Kreutz et al., 2015) han demostrado que las RT no son capaces de satisfacer esta nueva necesidad, ya que todos los componentes están integrados verticalmente, lo que se refiere a que todos los aspectos de su infraestructura, desde el hardware hasta el software, están estrechamente acoplados y controlados por un único proveedor o fabricante. Esto implica que el proveedor tiene un control total sobre los componentes y las políticas, por lo cual las opciones de configuración y personalización pueden estar limitadas a lo que se permita. Además, cualquier modificación o actualización en un componente puede afectar a otros elementos, creando una estructura compleja difícil de administrar. Por todo lo antes expuesto administrar este tipo de entornos presenta muchos desafíos debido a sus limitaciones, especialmente la falta de control central (Al-Haddad et al., 2021).

La *Calidad de Servicio* (QoS) se refiere a la capacidad de una red para proporcionar un mejor servicio a los usuarios, que pueden utilizar diferentes tecnologías tales como, *Frame Relay*, *Asynchronus Transfer Mode* (ATM), *Ethernet* e IP. El objetivo principal es proporcionar prioridad, incluyendo ancho de banda dedicado, latencia controlada (requeridas por cierto tráfico interactivo y en tiempo real) y mejora en parámetros cómo la pérdida de paquetes (Inc., 2003). El poder lograr ésto, es de vital importancia ya que puede garantizar, satisfacción del usuario, priorización de aplicaciones críticas, gestión de tráfico, soporte para servicios multimedia y aplicaciones exigentes y cumplimiento de las *Service Level Agreement* (SLA). La convergencia es un esfuerzo para integrar diferentes tipos de tráfico (voz, datos y video) en una sola topología y ésta es de gran importancia para el uso de nuevas aplicaciones y tecnologías. A pesar de esto, las RT pueden presentar desafíos en términos de gestión de QoS para garantizar la ade-

cuada transmisión de tráfico sensible en tiempo real, así como en la seguridad para proteger los diferentes tipos de tráfico que convergen en la misma infraestructura de comunicación. Las tareas de medición de la QoS suponen un mayor reto en este tipo de arquitecturas (Shah et al., 2020).

Para asegurar la calidad y confiabilidad del servicio de red se debe garantizar los SLA, el cual es un contrato formal entre un proveedor de servicios de red y un cliente que define los niveles de servicio que se deben cumplir. Esto se logra mediante la correcta gestión de los flujos de tráfico de los usuarios de forma eficiente para incrementar la QoS y respaldar éstos acuerdos de los clientes (Mohammadi y Javidan, 2021).

### 1.2. Antecedentes

El telediagnóstico, se refiere a la capacidad de realizar diagnósticos médicos de forma remota, este servicio está ganando cada vez más interés y atención en la comunidad médica y en la industria de la salud. El uso de TICs ha permitido que la telemedicina, incluido el telediagnóstico, superen las barreras geográficas y de tiempo, logrando de ésta manera mejorar el acceso a la atención médica especialmente, en áreas rurales o con escasez de recursos médicos.

En el contexto del telediagnóstico, la QoS sirve para garantizar una transmisión de datos confiable y oportuna entre los puntos de diagnóstico y las casas de salud. En este sentido, las SDN han surgido como un enfoque innovador y prometedor para mejorar la gestión y control de redes, lo que podría tener un impacto positivo en la QoS para el telediagnóstico.

Las SDN son una idea relativamente nueva en el ámbito de las redes, son un paradigma que separa el plano de control del plano de datos, permitiendo una mayor flexibilidad, programabilidad, automatización en la gestión de redes, mayor capacidad de adaptación a las necesidades de las aplicaciones, mejoras en el rendimiento, eficiencia, y posibilidad de implementar políticas de seguridad avanzadas, entre otras. En vez de tener el control de red distribuido en dispositivos individuales, como en las RT, SDN utiliza un controlador de red basado en aplicaciones de software, para facilitar la configuración y gestión de manera más eficiente y escalable. Además, SDN admite enfoques de enrutamiento dinámico del tráfico, lo que permite usar diferentes estándares y gestionar dinámicamente los flujos de tráfico, monitoreando el estado de la red y del flujo de datos (Jameel et al., 2022). Una de las grandes diferencias entre las RT y SDN, es el control que tiene los proveedores a la hora de configurar o personalizar un componente de red, ya que se debe limitar a lo permitido dependiendo de la marca, en cambio, para SDN los controladores SDN es de código abierto utiliza, por ejemplo, OpenDaylight (ODL) utiliza JAVA específicamente lo que implica que se puede realizar las configuraciones que se deseen sin

importar la marca del equipo con el que se está trabajando.

En la literatura, se han propuesto numerosas soluciones y enfoques para abordar los diferentes desafíos presentes en ésta tecnología y maximizar los beneficios de las SDN. Por ejemplo, se han desarrollado técnicas de optimización de rendimiento y escalabilidad de redes SDN, como la distribución eficiente de flujos, la gestión de la congestión y la asignación de recursos.

Para el caso de redes inalámbricas, (GANESAN et al., 2022), se propone un mecanismo de asignación dinámica de longitud de onda y ancho de banda (SD-TI-DWBA) basado en SDN. Este artículo realiza una simulación dónde los resultados muestran que al desplegar la arquitectura con el SD-TI-DWBA propuesto, redujo la latencia promedio de los paquetes para el streaming en directo *Tactile Internet - Peer to Peer* (TI-P2P) en un 20,62 % cuando el flujo de datos era del 100 %. También, se dedujo que la localización del *switch* de tráfico *OpenFlow* (OF) tiene un efecto considerable en el rendimiento del *jitter TI-P2P*. Por lo tanto, la arquitectura SD-TI-DWBA propuesta ofrece un mejor rendimiento del *jitter* cuando el flujo de tráfico es inferior al 100 % debido a los métodos de localización del tráfico del *switch OF*. Cuando el tráfico supera el 90 %, los escenarios SD-TI-DWBA comenzaron a perder paquetes. Además, las pérdidas de paquetes *Best Effort* (BE) del SD-TI-DWBA en el caso de tráficos elevados fueron diferentes en todos los escenarios, lo que indica que la arquitectura no afectará a la pérdida de paquetes. Por lo tanto, el trabajo concluyó que la arquitectura puede mejorar el rendimiento del sistema y las métricas de QoS en relación con, la latencia, el *jitter*, el rendimiento del sistema y la pérdida de paquetes.

Las métricas consideradas en (Kamboj et al., 2021) fueron: la latencia extremo a extremo, el *jitter* medio y el *throughput* medio, dónde se logró asegurar una mejor QoS en las aplicaciones. Se propuso una técnica para gestionar el ancho de banda en una red de *Internet of Things* (IoT) definida por software. Se formuló un problema de enrutamiento para encontrar rutas óptimas para diferentes flujos de tráfico (TCP o UDP) producidos mediante un *generador de tráfico de Internet distribuido* (D-ITG). En ésta simulación se demostró la eficacia del enfoque propuesto, consiguiendo una reducción de la latencia extremo a extremo del 26 %, 37 % con flujos TCP, 44 % con flujos UDP en comparación con los esquemas de referencia SDN con *Hierarchical Token Bucket* (HTB). Además, el enfoque consiguió un *throughput* medio superior en un 31 %, 51 % con flujos TCP, 43 % con flujos UDP en comparación con los esquemas de referencia utilizados.

En (Lumbantoruan et al., 2020) se propone una novedosa solución de enrutamiento extremo a extremo que combina la gestión de QoS y el enrutamiento multitrayecto utilizando el modelo de SDN para obtener una baja latencia para el tráfico de Internet. Los resultados de la simulación

muestran que la solución propuesta mejora el rendimiento del sistema y alivia la congestión entre los flujos en comparación con el *Enrutamiento más Corto de una Sola Ruta* (SSSP). Para garantizar la QoS del tráfico y los flujos de vídeo cuando hay algunos enlaces compartidos comunes en la ruta, se desplegó el mecanismo de cola para satisfacer los requisitos de cada tráfico y ésta función de QoS en la red generada por ellos puede funcionar bien incluso en entornos de red saturados.

El trabajo desarrollado en (Vanitchasatit y Sanguankotchakorn, 2022) se enfoca en la provisión de QoS de varias clases de tráfico en una red habilitada para SDN. Se propuso e implementó el esquema de control de QoS adaptativo basado en clases, en una red SDN con varios tipos de tráfico como, VoIP, *streaming* de vídeo y transferencia de archivos. La principal métrica de evaluación del rendimiento, es el número máximo de flujos de tráfico admisibles con QoS. Los resultados mostraron que el retardo aumentó drásticamente cuando la velocidad de datos supera los 12 Mbps en el caso del encaminamiento por una única ruta, mientras que el encaminamiento por rutas múltiples, puede mantener un nivel bajo de latencia, cuando la velocidad total de datos supera los 15 Mbps. Cuando la velocidad de datos de las fuentes de tráfico se sitúa alrededor de los 3 Mbps, la latencia del tráfico, el flujo de vídeo y el *Mejor Esfuerzo* (BE) es de 3,77ms, 2,18ms y 1,05ms, respectivamente. El rendimiento del flujo de vídeo es mejor que el del BE porque se configuró previamente que la prioridad del vídeo es la segunda, mientras que el BE no tiene prioridad. Cuando el total de paquetes procedentes de las fuentes de tráfico es de aproximadamente 34 % del ancho de banda máximo del enlace de la red, la solución propuesta puede seguir proporcionando una latencia baja (0,977 ms) para el tráfico.

La investigación desarrollada por (Medvetskyi et al., 2021) se trata de la simulación de un sistema de monitorización de QoE para *SDN basadas en la intención* (IBSDN), que mejora la QoE del usuario final y permite un uso más eficiente de los recursos. Los parámetros de red utilizados son: latencia y pérdida de paquetes. Se llevó a cabo una investigación para evaluar el rendimiento del sistema de monitorización propuesto generando tráfico de audio y vídeo en la red simulada. Como resultado, la solución proporciona una evaluación estable de la QoE global y la mantiene al nivel requerido basándose en los datos medidos en la ruta real de transmisión de datos. Al comparar los resultados obtenidos, se observó que el sistema de monitorización puede reducir el número de pérdidas de paquetes y, en general, mejorar la QoS en el caso del *streaming* de vídeo.

La llegada de las SDN elimina la mayoría de las limitaciones de las redes basadas en IP en términos de provisión de QoS. Este mecanismo, facilita la gestión y configuración dinámica para que los proveedores garanticen la QoS a sus clientes, especialmente en entornos dónde

se tienen múltiples servicios o productos. Estas capacidades, hacen de SDN un candidato adecuado para utilizarlo en arquitecturas de red con capacidad en telecirugía (Mohammadi y Javidan, 2018). En el análisis bibliográfico realizado, se han conseguido trabajos con el uso de éste enfoque para servicios de telemedicina, específicamente en el área de telecirugía.

En (Mohammadi et al., 2021) se describe una técnica de *Ingeniería de Tráfico* (TE) multiobjetivo denominada *SMOTE* para garantizar la *Calidad de Experiencia* (QoE) de las aplicaciones de telecirugía sobre SDN, de ésta manera el cirujano remoto puede ver los fotogramas de vídeo precisos y exactos transmitidos desde el lado del paciente. Este trabajo plantea la minimización así como, de la latencia extremo a extremo y la pérdida de paquetes. SMOTE utiliza el *Algoritmo Genético de Ordenación No Dominante* (NSGA-II, por sus siglas en inglés), un conocido algoritmo multiobjetivo, para resolver el modelo de optimización de cálculo de rutas óptimas entre el cirujano y el quirófano. Se realizaron simulaciones para evaluar el rendimiento de SMOTE en términos de QoE, entre los resultados se consiguió minimizar las pérdidas de paquetes de extremo a extremo, dado que recopila dinámicamente las estadísticas de ésta métrica para así calcular la ruta óptima teniendo en cuenta los enlaces con una mínima pérdida de paquetes. Ésta técnica supera considerablemente a *Short Path Route* (SPR) en términos de *Peak Signal Noise Ratio* (PSNR) y *Structural Similarity Index Measure* (SSIM), ya que minimiza tanto, la pérdida de paquetes como la latencia de la ruta óptima entre el cirujano y el robot. Por lo tanto, los resultados de las simulaciones confirmaron que SMOTE garantiza la QoE en telecirugía.

Por lo tanto, con ésto se demuestra la posibilidad de utilizar este tipo de enfoque ya que se han utilizado para diferentes aplicaciones y perspectivas.

### 1.3. Justificación

La QoS asegura que ciertas aplicaciones críticas reciban un trato preferencial para garantizar un rendimiento óptimo y una experiencia de usuario satisfactoria en la red utilizando diferentes tecnologías como, IP, LAN y WAN. Las métricas del rendimiento que afectan a la QoS son: ancho de banda, congestión de la red, latencia (retardo), *jitter* y tasa de error. Los investigadores han analizado los protocolos disponibles para las RT y han propuesto nuevas soluciones para admitir una amplia gama de aplicaciones, como voz, vídeo y transferencia de archivos. Sus soluciones fusionan protocolos tradicionales con las últimas tecnologías, como la virtualización, los últimos paradigmas de SDN y los mecanismos de fragmentación (Al-Haddad et al., 2021).

Debido a las limitaciones que presentan las RT, asegurar la QoS siempre será un problema.

Los sistemas comerciales, como Cisco, proporcionan una QoS global adecuada, considerando todos los tipos de tráfico y dan mayor prioridad a flujos de tráfico específicos. Sin embargo, estos sistemas aún presentan problemas de escalabilidad y congestión. Se han desarrollado numerosos estudios de investigación que sugieren diferentes enfoques para administrar la congestión. En teoría, la mayoría de esos métodos pueden manejar la entrega de paquetes cuando se necesita más ancho de banda del que puede admitir un enlace. Pero en la práctica, deben volver a evaluarse y validarse en nuevos sistemas de red extendidos, como SDN. Es posible que éstos puedan resolver varios problemas de QoS al proporcionar una visibilidad completa para recopilar y analizar el tráfico, y de ésta manera garantizar que los dispositivos sean programables (Al-Haddad et al., 2021).

Proporcionar QoS se está convirtiendo en un aspecto importante de cualquier arquitectura. Las nuevas aplicaciones y servicios tienen diferentes especificaciones. Por ejemplo, las redes de *Internet Industrial de las cosas* (IIoT), como las fábricas inteligentes, tienen requisitos estrictos, en la latencia ultrabaja, alta fiabilidad y configuración dinámica flexible (andTarik Taleb y Song, 2021).

En (Mohammadi et al., 2021), se presenta que al minimizar parámetros de QoS se logra mejorar la QoE del cirujano a distancia. Las métricas controladas fueron, la latencia extremo a extremo y la pérdida de paquetes ya que éstas afectan a la transmisión de vídeo.

Una de los grandes beneficios que se pueden aplicar dentro de las arquitecturas SDN es la asignación de ancho de banda entre servicios, en función de las prioridades de las aplicaciones, ya que ésta arquitectura permite políticas de TE avanzadas centralizadas (Troia et al., 2022).

Algunas de las aplicaciones multimedia son sensibles al retardo, mientras que otras son sensibles a la pérdida de paquetes. SDN es una tecnología emergente que proporciona flexibilidad a la red gracias a su naturaleza programable. Debido a la abstracción, permite la configuración dinámica de las políticas a nivel de aplicación y aborda las limitaciones de la arquitectura actual de Internet mejorando así el rendimiento de las aplicaciones multimedia (Kamboj et al., 2021).

En los últimos años, las innovaciones en las *Tecnologías de la Información* (TI) y las telecomunicaciones han reducido la importancia de la distancia geográfica. La telemedicina permite que los médicos puedan examinar y prescribir tratamientos a pacientes remotos. Puede aplicarse en diversas situaciones: teleenfermería, telefarmacia, telecardiología, telepsiquiatría, teleradiología, telecirugía, etc. El uso de éste, reduce los costos de atención, facilita la educación médica y hace que los distintos procedimientos sean más convenientes tanto para los médicos como para los pacientes (Mohammadi y Javidan, 2018).

Por su parte el telediagnóstico es una modalidad de atención a distancia, utilizando TIC para intercambiar datos, imágenes y resultados de pruebas médicas entre profesionales de la salud y pacientes ubicados en diferentes lugares geográficos. Este servicio se destaca por tener múltiples beneficios tanto para pacientes de cualquier índole social, cultural, económico sin importar su ubicación geográfica y para el doctor, además, el contar con ésto, no solo es acercarse a servicios innovadores sino solventar diferentes problemas que pueden surgir a la hora de querer realizar un diagnóstico tradicional.

Entre las soluciones que se pueden conseguir con este servicio se tienen:

- Eficiencia: puede agilizar el proceso de diagnóstico, reduciendo la necesidad de traslados y esperas para la obtención de resultados.
- Mejora de la precisión del diagnóstico: permite la colaboración y consulta entre profesionales de la salud, lo que puede llevar una mejor precisión del diagnóstico.
- Reducción de costos: puede reducir los costos asociados con los traslados y estadías de pacientes, así como con la repetición innecesaria de pruebas médicas.
- Ampliación de la cobertura: puede ampliar la cobertura de atención médica a poblaciones que enfrentan barreras socioeconómicas, culturales o de movilidad.

En cuanto a la tendencia de crecimiento en el uso de dispositivos como ordenadores portátiles y tablets en 2021. Los resultados de una encuesta realizada en Brasil, indicaron que el 29 % de los centros sanitarios utilizaban *tablets* y el 61 % portátiles, porcentajes superiores a los encontrados en 2019 (17 % y 48 %, respectivamente) (Center, 2021). Lo que implica que la tendencia para los próximos años, es que las casas de salud cuenten con los dispositivos tecnológicos para brindar servicios de telemedicina, pero la falta de optimización en el uso de redes, no permite que puedan sostener y asegurar este tipo de servicios haciendo más difícil el buen desarrollo de los mismos.

En la época de la pandemia se vio en la obligación de hacer uso de la telemedicina debido al aislamiento forzado por causa del COVID-19, dejando en evidencia la necesidad de buscar otras formas para brindar atención médica. Sin embargo, en las casas de salud de Cuenca, no se cuentan con éste tipo de servicios actualmente, limitando la posibilidad de ofrecer otra manera de diagnosticar.

En la sección 1.2 se mencionan como diferentes investigadores han sugerido soluciones para mejorar la QoS para diferentes áreas incluido telemedicina, pero específicamente para el área de telediagnóstico, no se ha investigado o propuesto una solución para mejorar la QoS utilizando la arquitectura SDN.

Para este trabajo de grado se tiene como caso de estudio la topología de red presente en la torre IV de la *Clínica Monte Sinaí* ubicada en Cuenca, Ecuador. Esta casa de salud, cuenta con una serie de consultorios de diferentes especialidades, además de una RT a la cual se le integrará la emulación de un escenario de telediagnóstico. También, se harán uso de diferentes métricas para evaluar la QoS de la red.

### 1.4. Alcance

El presente trabajo de grado tiene como objetivo evaluar métricas de QoS aplicando SDN para mejorar la transmisión de paquetes para telediagnóstico.

Para este trabajo, se plantea mejorar la QoS dentro de una red LAN para telediagnóstico. Para conseguir un escenario más realista, se hace uso de la topología de red de la Clínica Monte Sinaí (Torre IV) (**escenario real**), para ésta topología la caracterización del tráfico se definirá bajo la evaluación de métricas de QoS, dichas métricas serán determinadas a partir del análisis bibliográfico, como se muestran en (Medvetskyi et al., 2021), (Jahromi et al., 2018), (Jiawei et al., 2018) y serán medidas por medio de un analizador de red, mediante el envío de paquetes TCP y UDP con los tamaños característicos de los *Digital Imaging and Communications in Medicine* (DICOM). Seguidamente, se plantea la simulación basada en el escenario real con SDN (**escenario simulado**), dentro de éste nuevo escenario se volverán a enviar paquetes del mismo tamaño (enviados en el escenario real) para caracterizar la red con condiciones de tráfico similares a las que se sometió el escenario real y de ésta manera demostrar la hipótesis; se compararán los resultados de las métricas obtenidas en el escenario real vs. el escenario simulado.

El trabajo estudia la viabilidad de desarrollar un entorno SDN para mejorar la calidad de transmisión de paquetes para telediagnóstico, evaluando métricas de QoS para escenarios de telediagnóstico. ¿Es posible mejorar la QoS aplicando SDN para mejorar la transmisión de paquetes de telediagnóstico?.

Para el desarrollo del trabajo se empleará una metodología cuantitativa, donde se realizará un análisis comparativo de las parámetros medidos con el analizador de red tanto para el escenario simulado en MiniNet con la arquitectura SDN, y el escenario actual de la red de la Clínica Monte Sinaí.

## 1.5. Objetivos

### 1.5.1. Objetivo general

Evaluar métricas de calidad de servicio (QoS) aplicando redes definidas por software (SDN) para mejorar la transmisión de paquetes para telediagnóstico.

### 1.5.2. Objetivos específicos

1. Extraer las métricas óptimas de QoS para escenarios de telediagnóstico.
2. Definir el escenario y caracterización del tráfico de red del caso de estudio.
3. Diseñar el escenario SDN en el entorno de simulación.
4. Analizar las métricas de QoS obtenidas para el escenario con SDN y las obtenidas en el escenario actual de la red.

# Fundamentos Teóricos

En este capítulo se realiza una descripción de los fundamentos del área de la salud y tecnológicos que son necesarios para entender el contexto de éste trabajo de grado. Se define la salud digital y se explica la evolución de ésta, así como también la situación actual al nivel mundial y en Latinoamérica. Seguidamente se explica los conceptos involucrados con la telemedicina, específicamente para el área de telediagnóstico. Luego, se definen conceptos como, las redes tradicionales siguiendo con la QoS y las diferentes métricas utilizadas. Por último se explican todo los conceptos relacionados a las SDN.

## 2.1. Salud Digital

El término “salud digital” se usa ampliamente en diferentes campos, sin embargo, no hay una definición exacta. Las academias, instituciones científicas, la industria y las personas tienen diferentes perspectivas con respecto a éste término, lo que lleva a no tener una definición integral. En la revisión sistemática realizada por (Fatehi et al., 2020) se identificaron los siguientes elementos para referirse a la innovación en salud digital: salud electrónica, salud móvil, salud 2.0, telesalud y telemedicina, vigilancia de la salud pública, medicina personalizada, estrategias de promoción de la salud, sensores y dispositivos portátiles de seguimiento automático, genómica, imágenes médicas y sistemas de información. Se puede decir que es un amplio conjunto de prácticas y tecnologías. También se utilizan términos como “virtual”, “cibernético”, pero el término “digital”, es el que ha ganado más apoyo.

Este concepto abarca las formas de atención médica en línea (a través de Internet), el cual puede ser síncrono o asíncrono, síncrono se refiere a la transmisión de señales en tiempo real, que se destinan a la visualización o procesamiento inmediato, y asíncrono implica la obtención, almacenamiento y transmisión de datos del paciente para su posterior evaluación por parte de los profesionales de la salud (España, s.f.). Éste trabajo de grado simula una atención médica del tipo síncrono. Actualmente, los grandes avances en la tecnología son aplicados para apoyar la salud digital, entre estos se destaca, la *Inteligencia Artificial* (IA) y el aprendizaje automático como enfoques fundamentales combinados con las TICs y otras tecnologías utilizadas por organizaciones de salud, médicos, pacientes y consumidores, con el propósito de mejorar el estado de salud de los pacientes (Fatehi et al., 2020), (España, s.f.), (Thompson, 2021).

Al realizar un enfoque de la salud desde una perspectiva geográfica se debe considerar no sólo dónde se aplican prácticas para la salud, sino qué sucede en los lugares, que puedan afectar

a éste. Sin embargo, pocos estudios han intentado explorar Internet como otro espacio de salud, aunque algunos estudios recientes han mencionado la importancia de las plataformas en línea, como las comunidades de apoyo al paciente, para promocionar la atención médica (Thompson, 2021).

El término de salud digital fue introducido por (Frank, 2000) en el año 2000, a partir de ese momento sigue evolucionando. Incluye en gran medida medios y aplicaciones en línea para mejorar el contenido de salud, comercio y la conectividad. Para el 2018, la *Organización Mundial de la Salud* (OMS) publicó una clasificación detallada que describe decenas de aspectos para éste término (Mathews et al., 2019). En el 2019, la *Asociación Médica Americana* (AMA) declaró que las empresas han invertido miles de millones de dólares en nuevos proyectos para éste tema. La OMS enfatiza que la salud digital puede ser útil para lograr los Objetivos de Desarrollo Sostenible al hacer que los servicios de salud y bienestar estén disponibles para todo el mundo (Fatehi et al., 2020).

### 2.2. Evolución de la salud digital

Para los años 90 la salud digital era un término que se utilizó principalmente para referirse a la digitalización de información y bibliotecas de salud. Con la expansión de Internet en todo el mundo a partir del año 2000, éste concepto cambió. Posteriormente, con el desarrollo de las ciencias de la computación, y sus aplicaciones en el cuidado de la salud, también se consideraron algunos conceptos nuevos como parte de la salud digital. A pesar de que aún el término y su clasificación es muy ambigua, existió la necesidad de consolidar éste concepto para su uso en la investigación, la política y la práctica (Fatehi et al., 2020).

La FDA (*Food and Drug Administration*) de EE. UU. considera una amplia gama de tecnologías como salud digital: salud móvil, dispositivos portátiles, telesalud y telemedicina, tecnologías de la información en salud y medicina personalizada. Las soluciones sanitarias digitales están creciendo rápidamente; para la fecha hay más de 300.000 aplicaciones de salud.

Cómo se mencionó anteriormente las inversiones a nivel mundial en atención médica digital son enormes, con casi 6.000 millones de dólares de financiación para el 2017, frente a los 4.400 millones del 2016. Solo en el caso de las aplicaciones móviles de salud se agregan 200 diariamente, lo que lleva a un panorama cada vez más amplio y confuso por donde pueden navegar todas las partes interesadas en el cuidado de la salud: pacientes, proveedores, industria y reguladores (Mathews et al., 2019). Algunos ejemplos de éstas aplicaciones, es una plataforma en línea que utiliza IA para diagnosticar a pacientes en áreas rurales de Brasil, la misma que permite subir imágenes médicas desde clínicas remotas y se puede obtener un

diagnóstico preliminar en 24 horas, a un costo de $4 por examen (Elizabeth y Andrea, 2019).

### 2.3. Fundamentos teóricos de la salud digital

La telemedicina se presenta como una práctica esencial en términos de salud, tecnología, cultura y sociedad, con el objetivo de mejorar el acceso a la atención médica, la calidad de la atención y la eficiencia organizativa. Se define como el uso de tecnologías de la información y la comunicación para proporcionar servicios de salud a distancia en áreas como promoción, prevención, diagnóstico, tratamiento y rehabilitación. Los profesionales de la salud utilizan estas tecnologías para intercambiar datos y así facilitar el acceso y la oportunidad en la prestación de servicios médicos.

Inicialmente, la telemedicina fue concebida y desarrollada con el propósito de acercar los servicios de salud a poblaciones que viven en áreas remotas y con limitado acceso a recursos médicos. Por lo tanto, las primeras definiciones de este concepto hacían hincapié en la importancia de la distancia. Sin embargo, la pandemia de COVID-19 ha actuado como un impulsor de la telemedicina, destacando su uso, importancia y utilidad como estrategia para mantener la atención médica sin tener que estar en la casa de salud (Marisa, 2022).

El hecho de que el paciente no tenga que movilizarse hasta una casa de salud o que no tenga que ausentarse del trabajo, puede significar mucho. La acción de acudir a la consulta y estar sentado en espacios reducidos con otras personas, puede provocar infecciones lo que es especialmente arriesgado para pacientes que tienen problemas de salud crónicos o sistemas inmunes debilitados. Estos problemas se pueden evitar gracias a la telemedicina que permite que tanto el paciente como el médico reciban y brinden, respectivamente, atención médica de manera conveniente y segura. Desde el punto de vista económico, los proveedores de telemedicina pueden tener costos de entrada más bajos y los médicos pueden encontrar que la telemedicina aumenta sus ingresos al permitirles ver a más pacientes.

La videoconferencia inicialmente atrajo a muchos proveedores a la telemedicina, pero la nueva ola de tecnologías en ésta área ofrece mucho más. Los avances más recientes incluyen IA para ayudar a los médicos a trabajar de manera más eficiente. Esta tecnología permite a los pacientes usar dispositivos portátiles modernos, entre ellos los *wereables*, que son dispositivos tecnológicos que el paciente utiliza, éstos recopilan y procesan información sobre la salud del paciente, desde la monitorización de diversas constantes hasta la presentación de los resultados procesados y otros recursos para monitorear de forma remota (España, s.f.). También se pueden usar robots para brindar tratamiento especializado en áreas en las que nunca antes se había recibido, como por ejemplo, *Angel Robot*, es un robot de la empresa *Meridian*

*Medical Network Corp* el cual emplea la tecnología de reconocimiento facial para identificar y comunicarse con los pacientes, con el objetivo de referirlos a especialistas o prescribirles medicamentos (Elizabeth y Andrea, 2019).

Para las zonas remotas (rurales) las tecnologías de telemedicina es muy prometedora. El efecto más significativo se ha visto en países donde hay muy pocos casas de salud. Algunas pueden ofrecer citas virtuales con médicos a través de videoconferencias. Cuando las visitas presenciales no son necesarias, éste tipo de servicio remoto le permiten continuar el tratamiento con su médico. Como parte de los servicios médicos, algunas corporaciones importantes tienen acceso a salas de salud automáticas (a et al., 2021).

Dentro de la telemedicina existen diferentes manera de clasificarla según la *Organización Panamericana de la Salud* (OPS) con el aval de la OMS, éstas son: en el tiempo, en las especialidades y en el tipo de aplicación médica. Según el tiempo, se refiere al momento en que se realiza la intervención médica a distancia y la comunicación entre el proveedor del servicio y el cliente: tiempo diferido y tiempo real. En cuanto al tipo de servicio se tiene: teleconsulta, telediagnóstico, telecuidado (teleatención), telemetría (telemedida), teleeducación, teleadministración, teleterapia (telepsiquiatría, telefisioterapia, teleoncología, teleprescripción) y telefarmacia entre otras. Y por último, en la clasificación por especialidades se tiene: telerradiología, telepatología, telecariología, teleORL, teleendoscopia, teledermatología, teleoftalmología y telecirugía (OPS/OMS, 2000).

En Ecuador, la telemedicina es un campo que aún requiere una mayor investigación y desarrollo. Aunque se han realizado trabajos como, el programa nacional de Telemedicina/Telesalud, no se ha establecido una política pública que respalde e impulse la implementación e investigación de la telemedicina en Ecuador. Uno de los principales desafíos identificados es la conectividad, debido a las condiciones geográficas del país, que han limitado el acceso a Internet en los establecimientos de salud rurales, impidiendo su inclusión en la implementación de estas tecnologías. Es necesario llevar a cabo una amplia labor, especialmente en la capacitación de profesionales de la salud y en la adquisición del equipamiento necesario para garantizar el correcto funcionamiento de la telemedicina. Además, es importante promover el uso de las TICs en beneficio de la salud de la población (Puertas, 2019).

Este trabajo de grado, busca mejorar la calidad de las redes de una casa de salud dónde el tráfico generado tenga características especiales que sean aplicados en escenarios de telemedicina. Por lo tanto, de la clasificación antes mencionada, se ha identificado al telediagnóstico como el más adecuado, como usuario de varias características de la telemedicina, por lo que necesita transferir, paquetes de estudio (DICOM), la evolución de la historia clínica electrónica,

videoconferencia, de tal forma que el escenario propuesto cubra varias aristas de atención de telemedicina.

El telediagnóstico se define como, la determinación de la naturaleza de la enfermedad desde un lugar alejado del paciente, basado en métodos de transmisión remota de datos. Ésto le permite obtener atención médica sin salir de su casa. Al ahorrar tiempo de viaje, simplifica el diagnóstico y tratamiento de algunas enfermedades crónicas. El telediagnóstico tradicional por teléfono y videoconferencia se ha desarrollado durante mucho tiempo, sus deficiencias también son muy obvias. Los médicos solo pueden obtener información limitada a través de dispositivos electrónicos tradicionales. Y debido a que es tan diferente de la forma habitual de diagnóstico y tratamiento, es fácil que los médicos presten menos atención e influyan más en los resultados de la consulta (Fu, 2020).

En los últimos años, se han desarrollado muchas propuestas de telediagnóstico, dos de las cuales son:

- Aplicación MTS para telediagnóstico: el sistema de telemedicina móvil (*Mobile Telemedicine System*, por sus siglas en inglés MTS) permite la recopilación de datos de diferentes dispositivos, la transmisión y envío al sistema de seguridad social y la transferencia al proveedor de atención médica a través de la red 3G/4G. Se puede acceder al MTS desde cualquier zona geográfica utilizando la red de comunicaciones 3G/4G y los servicios en línea para el diagnóstico de enfermedades crónicas como diabetes, enfermedades de la piel y electrocardiograma (ECG).
- Enfoque de aprendizaje automático para telediagnóstico: se ha demostrado que el aprendizaje automático, incluidas las redes neuronales profundas, es útil para diagnosticar diversas enfermedades (Qazi et al., 2019).

En términos generales, el funcionamiento del telediagnóstico implica los siguientes pasos como se muestra en la figura 2.1:

1. **Recopilación de datos:** el paciente o un profesional de la salud en su ubicación local recopila datos relevantes sobre la condición médica, como imágenes médicas, datos biométricos, síntomas o historial médico.
2. **Transmisión de datos:** los datos recopilados se envían de forma segura a través de una red de comunicación, como Internet, a un profesional médico o especialista ubicado en otro lugar.

3. **Análisis y evaluación:** el profesional médico recibe los datos y los analiza para realizar un diagnóstico. Puede utilizar herramientas de software especializadas para analizar imágenes médicas, realizar pruebas de laboratorio virtuales o evaluar los síntomas informados por el paciente.

4. **Comunicación de resultados:** una vez que se ha realizado el diagnóstico, el profesional médico se comunica con el paciente para proporcionar los resultados y discutir las recomendaciones de tratamiento. Esto puede realizarse a través de videoconferencia, llamada telefónica u otros medios de comunicación.

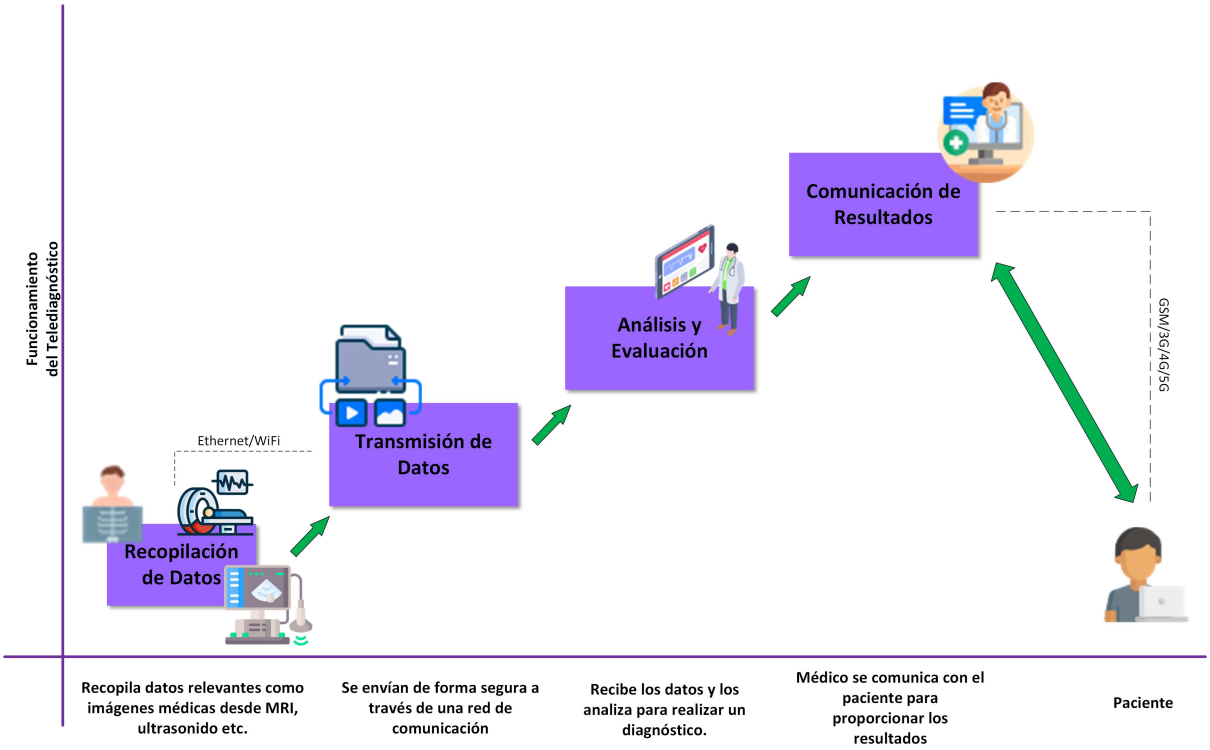



Figura 2.1: Funcionamiento del Telediagnóstico

## 2.4. Tecnologías de la Información y comunicación (TIC) aplicadas a la salud digital

Desde el 1950, comenzó el uso de aplicaciones en el cuidado de la salud, la cual se podría denominar la era de las TI en la salud. Para ésta época la infraestructura de telecomunicaciones desplegada era extremadamente limitada para los estándares actuales. A mediados de los 60 una serie de avances en las TI dieron lugar a un salto evolutivo, primero en forma de hardware y software, luego en el campo de las telecomunicaciones. El surgimiento de los miniordenadores y los ordenadores personales, la distribución de redes de telecomunicaciones con el creciente ancho de banda, la mayor abstracción y el uso asociado con los lenguajes

modernos de software y la mercantilización de las TICs, aceleró la expansión de las TICs sanitarias. Las TICs se implementaron en hospitales y luego en servicios de salud pública. Los grandes sistemas integrados coexistieron con los más avanzados en los hospitales para satisfacer las necesidades de los departamentos y de toda la organización, y los sistemas de gestión de consultas se desarrollaron e implementaron en las casas de salud más pequeñas.

A partir del año 2000 las TICs sanitarias se centraban en la organización de atención médica, específicamente en el proveedor y en el uso corporativo: se consideró más atención al tratamiento de la logística y la efectividad que las necesidades clínicas y de los pacientes.

Posteriormente, el impulso y los factores que promovieron el uso de las TICs en el campo de la sanidad se originaron tanto por la demanda como por la oferta.

La necesidad de un flujo de información en todo el sistema, llevó a la búsqueda de la interoperabilidad de la atención médica, y los gobiernos de todo el mundo se convirtieron en protagonistas y patrocinadores de la salud electrónica. La cibersalud fue ampliamente reconocida como el cambio de paradigma que justificaba este nuevo término.

En el Ecuador, sin embargo a partir del Primer Plan de Telemedicina se buscó establecer parámetros de atención, especialmente de teleasistencia por medio de, una red de videoconferencia para poder establecer una segunda opinión o la opinión del especialista, dando atención en zonas rurales (López-Pulles et al., s.f.). Dicho plan tuvo una etapa en el año 2013 el cual se expandió al programa de telemedicina a nivel nacional. Éstos intentos no tuvieron el éxito esperado; sin embargo, se consiguió empezar discusiones sobre acceso a la tecnología desde las casas de salud del gobierno y la interoperabilidad y uso de historial clínico unificado, en dichas instituciones.

Por último, la atención se centró en el sistema de prestación de atención médica, no sólo en las organizaciones individuales. Los pacientes comenzaron a tener acceso a (algunos de) los registros del proveedor, pero la mayoría de los datos aún estaban vinculados y controlados por el proveedor (Rowlands, 2020).

Para la época del 2020 en adelante, tecnologías como, el análisis de datos, la IA y el aprendizaje automático, la robótica y los servicios basados en la nube, así como los avances en campos como, la nanotecnología y la neurociencia, han madurado. La expansión del *Internet de las cosas* (IoT), las aplicaciones de salud, los dispositivos “siempre conectados” y las interfaces inmersivas, como la realidad aumentada y virtual, hipersuperficies y lingüística de la salud, están haciendo que la captura de datos médicos de calidad sea fácil y ubicua, lo que implica que se están generando mayores expectativas tanto en los consumidores como en los profesionales de la salud.

### 2.5. Contexto y estado actual de la salud digital

La era de salud digital está centrada en los ciudadanos, no solo se trata del sistema sanitario. Con el crecimiento exponencial de las TICs en la sociedad y economía digital, los ciudadanos exigen el control sobre su salud y bienestar, y esperan un servicio a la altura de sus necesidades como cualquier otro servicio: digitalmente, con flujos de trabajo integrados en sus pautas de vida y no en las pautas de conveniencia de los proveedores de servicios sanitarios.

La proliferación de las TICs cada vez más sofisticadas en la vida de las personas permitirá que la atención médica se brinde de manera más integral cuándo, dónde y cómo los pacientes la necesiten, especialmente en la atención primaria. Ésto promoverá la prevención, la detección y la intervención temprana, así como en la salud de precisión, lo que permitirá a los ciudadanos decidir cómo y dónde se asimilan y utilizan sus datos (Rowlands, 2020).

La OPS ha lanzado una iniciativa en América Latina y el Caribe llamada IS4H (*Information Systems for Health*) para mejorar los sistemas de información de salud en la región. Como parte de esta iniciativa, se ha desarrollado un modelo de evaluación de madurez que permite evaluar el estado de los sistemas de información de salud en cada país. Esta herramienta analiza la organización y gestión de la información, verifica si las decisiones en salud pública se basan en datos, evalúa la gobernanza de la información y determina la implementación de un sistema de información de salud (Elizabeth y Andrea, 2019).

El Bien Público Regional (BPR) “Transformación Digital en Salud para Mitigar los Efectos de COVID-19 en América Latina y el Caribe” (RG-T3769) busca fortalecer la capacidad de los países de la región para hacer frente a los impactos de la pandemia de COVID-19 a través de la promoción de la transformación digital en el sector de la salud.

Entre los objetivos principales que tiene el proyecto regional, como tercer componente se espera, establecer directrices para el desarrollo sostenible de la telesalud en la región, y ésto se desarrollará en un plazo de 3 años. Entre los países miembros del BPR son Argentina, Colombia, El Salvador, Guatemala, Paraguay, Suriname y Uruguay (Jennifer et al., 2022).

El Ministerio de Salud de Ecuador, basado en el documento “Consenso de recomendaciones de cuidados paliativos en la pandemia por SARS-CoV-2/COVID-19”, enfatiza la importancia de garantizar la continuidad de atención para pacientes con necesidades paliativas. Se sugiere la coordinación de referencia al equipo de cuidados paliativos de atención primaria o la implementación de seguimiento a través de telemedicina cuando sea posible. El documento menciona diferentes medios apropiados para estos propósitos, como llamadas telefónicas, plataformas de telemedicina basadas en video o plataformas de video chat comerciales, siempre

respetando la privacidad del paciente (Marisa, 2022).

## 2.6. Formatos de Imágenes Médicas

### 2.6.1. PACS

Un *Sistema de Archivo y Comunicación de Imágenes* (PACS) es una herramienta computarizada utilizada principalmente en organizaciones de atención médica que reemplaza la función de la película radiológica tradicional: adquisición, almacenamiento, transmisión y visualización de imágenes médicas e informes en formato digital. Las casas de salud ya cuenta con dicha herramienta por lo que el uso de las películas ha quedado muy limitado (Strickland, 2015).

El uso de PACS elimina la necesidad de archivar y almacenar físicamente, recuperar y enviar información sensible como, videos e informes. En cambio, los registros médicos y las imágenes se pueden almacenar de forma segura en servidores externos y se puede acceder a ellas desde cualquier parte del mundo utilizando el software PACS, las estaciones de trabajo y dispositivos móviles (Charles, s.f.).

### 2.6.2. DICOM

*Digital Imaging and Communications in Medicine* (DICOM) es la norma internacional para imágenes médicas e información relacionada. Define los formatos de las imágenes médicas que pueden intercambiarse con los datos y la calidad necesarias para el uso clínico.

DICOM se utiliza en la mayoría de los equipo de rayos X, cardiovasculares y de radioterapia (rayos X, tomografía computarizada, resonancia magnética, ultrasonidos, etc.), y cada vez más en otros campos médicos como la oftalmología y la odontología. Con cientos de miles de dispositivos de imagen médica en uso, DICOM es uno de los estándares de mensajería de atención médica más utilizados en el mundo. En la actualidad, se utilizan miles de millones de imágenes DICOM en la práctica clínica.

Desde su primera publicación en 1993, DICOM ha revolucionado la práctica de la radiología al permitir el reemplazo de la película de rayos X con un flujo de trabajo completamente digital. Justo cuando Internet se convirtió en la plataforma para nuevas aplicaciones de información para el consumidor, permitió aplicaciones de imágenes médicas de vanguardia que “cambiaron la faz de la medicina clínica”. Desde los departamentos de emergencia hasta las pruebas de esfuerzo cardíaco y la detección del cáncer de mama, es el estándar que hace que las imágenes médicas sean útiles tanto para los médicos como para los pacientes. Está reconocido

por la *Organización Internacional de Normalización* (ISO) como la norma **ISO-12052** (“About DICOM: Overview”, s.f.).

La composición del estándar DICOM incluye los siguientes elementos principales:

1. **Protocolo de comunicación:** DICOM define un protocolo de comunicación basado en TCP/IP que permite la transferencia segura y confiable de datos médicos entre diferentes dispositivos y sistemas.

2. **Formato de archivo de imagen:** DICOM establece un formato de archivo específico para las imágenes médicas, como tomografías computarizadas (TC), resonancias magnéticas (RM), radiografías, ecografías, entre otros. Estos archivos contienen tanto la imagen como los metadatos asociados, como información del paciente, detalles de la adquisición de la imagen y parámetros de configuración del equipo médico.

3. **Estructura de datos:** DICOM define una estructura de datos jerárquica y estandarizada para organizar y almacenar la información médica. Esto incluye la definición de elementos de datos específicos y sus atributos, como identificadores de pacientes, fechas, descripciones de procedimientos, resultados de pruebas, etc.

4. **Servicios de DICOM:** DICOM especifica una serie de servicios para el intercambio de datos médicos, como la recuperación y el envío de imágenes, consultas de información de pacientes, notificaciones de eventos y almacenamiento a largo plazo de datos médicos.

5. **Integración con sistemas de información:** DICOM puede integrarse con sistemas de información médica más amplios, como sistemas de gestión de imágenes y archivos médicos (PACS) y sistemas de información hospitalaria (HIS), para permitir una gestión más eficiente y accesible de la información médica.

### 2.7. Redes Tradicionales

Una RT está compuesto por diversos dispositivos de red, como *routers*, *switches* y puntos de acceso, cada uno con su propia lógica de control local para establecer conexiones y gestionar los paquetes (Jackisch, 2022). En este enfoque convencional, todos los planos de datos y de control se integran en una sola unidad física, lo que conlleva a compartir su capacidad y aumentar la carga de tráfico, así como la carga en la CPU y la memoria en dos procesos distintos (Haji et al., 2021).

Tanto la aplicación que inicia el flujo como el operador carecen de un control completo sobre el mismo. La complejidad y rigidez inherentes de las RT dificultan su capacidad para satisfacer los requisitos de diversas partes interesadas y aplicaciones. Los ingenieros se enfrentan al desafío de gestionar tareas complicadas. Para aplicar políticas, se deben configurar manualmente cada dispositivo de múltiples proveedores utilizando herramientas e interfaces específicas de cada proveedor, lo cual puede ser un proceso lento.

En el plano de control, las RT siguen un enfoque distribuido donde cada dispositivo funciona de manera independiente mediante protocolos como ARP, STP, OSPF, EIGRP, BGP y otros. Aunque estos dispositivos se conectan entre sí, no hay una máquina centralizada encargada de administrar y coordinar toda la red. Éstas se basan principalmente en hardware para su funcionamiento (Haji et al., 2021).

## 2.8. Calidad de Servicio (QoS)

Según la *Unión Internacional de Telecomunicaciones* (UIT), la QoS se define como: “el conjunto de características de un servicio de telecomunicaciones que influyen en su capacidad para satisfacer las necesidades declaradas e implícitas del usuario del servicio”(Letaifa, 2019). El objetivo principal es dar prioridad con respecto a ciertos parámetros, entre los que se incluyen: ancho de banda, latencia, *jitter* y pérdida de paquetes. El modelo de *Servicio Integrado* (IntServ) y el modelo de *Servicio Diferenciado* (DiffServ) se consideran los métodos convencionales.

### 2.8.1. Latencia extremo a extremo

La latencia extremo a extremo o latencia unidireccional se refiere al tiempo que tarda un paquete en transmitirse desde el origen hasta el destino para cada paquete de datos entregado con éxito (Mohammadi y Javidan, 2021), (HUSSAINI et al., 2017). Es un término común en el monitoreo de redes IP y se diferencia del tiempo de ida y vuelta (RTT) en que solo se mide la ruta en una dirección desde el origen hasta el destino.

### 2.8.2. Jitter

*Jitter* se define como la diferencia en los tiempos de viaje de los paquetes consecutivos en una conexión de extremo a extremo. En condiciones ideales, todos los paquetes deberían experimentar el mismo tiempo de viaje. Sin embargo, debido a factores como la congestión del tráfico, las variaciones en el tiempo de procesamiento en los nodos e incluso los cambios

en la ruta, los paquetes pueden experimentar fluctuaciones en sus tiempos de viaje, lo que se conoce como *jitter*. Esta variabilidad en los tiempos de viaje puede afectar la calidad de la transmisión y dar lugar a errores en la interpretación de los datos recibidos. Un *jitter* negativo se conoce como fenómeno de agrupamiento de paquetes, mientras que un *jitter* positivo se conoce como dispersión de paquetes (Rizo-Dominguez et al., 2010).

El *jitter* es una métrica de QoS que suele ser importante en aplicaciones multimedia. Un *jitter* elevado tiene efectos adversos en la calidad de la voz y el vídeo. Obviamente, un largo tiempo de recuperación de un fallo de enlace aumenta el **jitter** (Mohammadi y Javidan, 2021).

### 2.8.3. Packet Loss Ratio

La tasa de pérdida de paquetes se refiere a la proporción entre el número de paquetes perdidos en la prueba y el grupo de datos enviado. El método de cálculo es: $[(\text{mensaje de entrada - mensaje de salida}) \ / \ \text{mensaje de entrada}] * 100\,\%$.

Sin embargo, en casos prácticos, es común realizar esta estimación antes de la implementación completa de la red, por lo que es necesario calcular con precisión la tasa de pérdida de paquetes utilizando ciertos parámetros de la red. Normalmente, se utiliza un método de cálculo lógico directo para estimar la tasa de pérdida de paquetes en toda la red (Rizo-Dominguez et al., 2010).

### 2.8.4. Throughput

Es la tasa con la cual la red pudo enviar datos desde la fuente hasta el destino. El *throughput* generalmente se mide en kilobits por segundo (Kbps) (HUSSAINI et al., 2017).

La razón principal por la que se mide el rendimiento es determinar la parte alcanzable de la capacidad de transmisión, influenciada por la latencia y la pérdida de paquetes, para diferentes servicios. Desde el punto de vista del usuario, el rendimiento es una medida clave del desempeño para un servicio dedicado percibido por el usuario a nivel de aplicación (ETSI, s.f.).

El rendimiento se puede calcular utilizando la ecuación 2.1,

$$\text{througput} = \text{Cantidad de datos transferidos}/\text{Duración} \tag{2.1}$$

### 2.8.5. Ancho de Banda

En términos prácticos, el ancho de banda se refiere la cantidad de información que puede ser transmitida a través de cualquier tipo de conexión en un tiempo determinado. Por lo general,

el ancho de banda se mide en bits por segundo (bps) (Castro, 2009).

### 2.9. Software Define Network (SDN)

SDN proporciona una programación más eficiente de los recursos de *entrada/salida* (E/S) utilizadas para la transmisión de datos a través de una red. Permite a los ingenieros aplicar y desarrollar configuraciones mediante la programación. La estructura consiste en la separación del plano de control que decide cómo reenviar el tráfico, del plano de datos, el plano que realmente reenvía el tráfico, como se ve en la figura 2.2. Por lo tanto, los dispositivos se convierten en simples conmutadores. Se proporciona una *Interfaz de Programación de Aplicaciones* (API) a los operadores y aplicaciones, lo que les permite aplicar el control sobre las reglas de reenvío de paquetes en los dispositivos.

SDN separa el plano de aplicación, el plano de datos y el plano de control, lo que simplifica el control y la gestión de la red. Consta de dos componentes principales: el controlador y los *switches*. El plano de aplicación SDN es una capa compuesta por diversas aplicaciones que se comunican a través de API *northbound* (es una interfaz para permitir la comunicación entre el controlador de la red y las aplicaciones externas), es responsable de la gestión y de funcio-

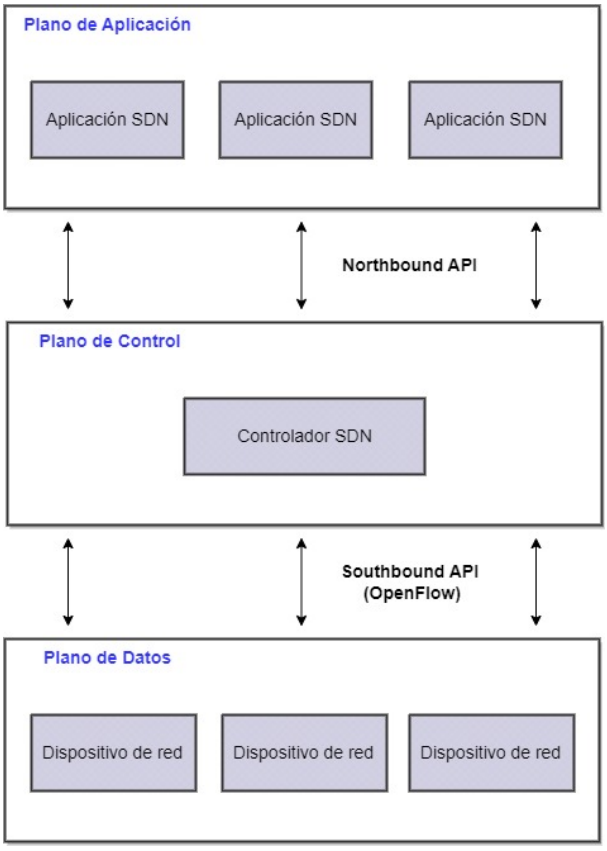



Figura 2.2: Modelo general de SDN

nalidades de información como la supervisión o la seguridad (Letaifa, 2019). El controlador es responsable de administrar toda la infraestructura de comunicación, controla los elementos de reenvío mediante la API de *southbound* (interfaces que sirven para establecer la comunicación entre el controlador y los dispositivos físicos), en función de reglas de administración de red. *OpenFlow* (OF) es uno de los protocolos *southbound* más populares y ampliamente utilizados por los controladores. Los *switches OF* son conmutadores que puede admitir el protocolo OpenFlow los cuales tienen una o más tablas de flujo, éstas contienen entradas que proporcionan la ruta de reenvío para los datos. Cada tabla tiene un número de registro que contiene un conjunto de campos tales como, la IP de origen, la IP de destino, el encabezado de paquete, una acción y uno o más contadores. También incluye otras características estadísticas como el número de bytes, la duración, etc. Cada entrada en esta tabla de procesos se completa con una regla de reenvío recibida del controlador. La capa de aplicación incluye aplicaciones como, seguridad, TE, gestión de QoS, tolerancia a fallas, etc. (Mohammadi y Javidan, 2021). A diferencia de las RT, donde todo el sistema debe reconfigurarse para las actualizaciones, SDN permite una configuración dinámica y eficiente desde el punto de vista programático, es decir, solo se requieren actualizaciones de software, lo que hace que la actualización sea más conveniente y reduce los costos totales, con el fin de mejorar el rendimiento y la supervisión. SDN tiene una gran demanda porque es flexible y se puede programar con cualquier lenguaje de programación de alto nivel para servir a los dispositivos y usuarios. Le permite adaptar la configuración en función del entorno operativo para mejorar el rendimiento general y detectar interrupciones en la misma (Rawat et al., 2017).

El nuevo concepto de red programable es la solución para la gestión integral. Aunque el número de usuarios que se conectan es cada vez mayor, no todos ellos necesitan la misma cantidad de recursos. La implantación de una QoS adaptativa mejorará la gestión de los recursos para obtener una mayor eficiencia y un mejor rendimiento (Vanitchasatit y Sanguankotchakorn, 2022).

Este enfoque es capaz de proporcionar servicios con programabilidad en tiempo real. La división entre el plano de datos y el plano de control tiene como resultado una mejora en la prestación de QoS para las diferentes aplicaciones existentes. La QoS puede aprovecharse de éste tipo de tecnología de varias formas, como mecanismos de enrutamiento, monitorización de red, reserva de recursos, gestión de colas y mecanismos de programación (andTarik Taleb y Song, 2021).

Algunas de las implementaciones de conmutadores virtuales son *Indigo*, *Open vSwitch* y *Pantou*. Los conmutadores virtuales soportan todas las funciones del protocolo SDN, mientras

que los conmutadores físicos carecen de flexibilidad y de funciones completas. La principal responsabilidad de los conmutadores en el plano de datos es reenviar, eliminar y modificar paquetes basándose en las reglas de flujo recibidas del controlador en el plano de control (Selvi y Thamilselvan, 2020).

El controlador también proporciona funcionalidades como enrutamiento, almacenamiento de topología, configuración de dispositivos, información de estado, etc. Los distintos controladores disponibles son *NOX*, *POX*, *Ryu*, *OpenDayLight*, *FloodLight*, *Beacon*, etc. El controlador proporciona la visión global a través de la cual se puede realizar la clasificación del tráfico. Éste busca identificar las reglas altamente relacionadas con cada paquete, haciendo uso de los bits pertinentes para seleccionar de manera óptima las reglas adecuadas. De esta manera, se logra una correspondencia efectiva entre las reglas y los paquetes entrantes (Selvi y Thamilselvan, 2020).

### 2.9.1. Protocolo OpenFlow

El protocolo OpenFlow es esencial para la implementación de SDN, ya que permite la separación del control y el reenvío de datos. Con el propósito de fomentar el desarrollo de este tipo de enfoque y establecer un estándar unificado, se fundó la *Open Networking Foundation* (ONF) en marzo de 2011. La ONF se dedica principalmente a promover SDN basado en el protocolo OF como el nuevo estándar de redes. Sus principales logros de investigación incluyen la definición de la arquitectura básica, el estándar y el protocolo de configuración y gestión de OF. Desde la primera versión de la especificación en octubre de 2009, la ONF ha lanzado varias versiones, como 1.1, 1.2, 1.3 y la más reciente, 1.4, en octubre de 2013 (Chena et al., 2019).

El protocolo se compone principalmente de puertos, tabla de flujos, canal de comunicación y estructura de datos. Su proceso de agrupación de datos se basa en el siguiente principio: después de recibir un paquete de datos, el *switch* SDN busca coincidencias en la tabla de flujos local. Los datos pasan por varias tablas, conocido como canalización. Cada entrada en la tabla tiene tres partes: coincidencia (se basa en el puerto de entrada, el campo de encabezado y la información previa), conteo (registra las coincidencias exitosas) y operaciones (como la salida al puerto, la encapsulación hacia el controlador o el descarte). Si un grupo de datos coincide con una entrada en la tabla de flujos, se actualiza el contador y se realiza la operación correspondiente. Si no hay coincidencia, se envía un mensaje o resumen al controlador para que decida el puerto de reenvío (Chena et al., 2019).

OF consta de tres tipos de comunicación: comunicación controlador-*switch*, comunicación asincrónica y comunicación simétrica. La comunicación controlador-*switch* se encarga de de-

tectar características, configurar y programar el *switch*, así como obtener información. La comunicación asincrónica es iniciada por el *switch* compatible con OF sin necesidad de una solicitud previa por parte del controlador. Se utiliza para informar al controlador sobre la llegada de paquetes, cambios de estado en el *switch* y errores. Y por último, la comunicación simétrica consiste en el intercambio de mensajes sin una solicitud específica de ninguna de las partes, es decir, tanto el *switch* como el controlador pueden iniciar la comunicación de forma independiente (Braun y Menth, 2014).

Es posible medir la QoS de manera cuantitativa y cualitativamente. Entre las ventajas de utilizar el paradigma SDN para garantizar la QoS se puede nombrar:

- El controlador tiene una visión global de toda la red.
- El conjunto de políticas y clases de flujo no está restringido, mientras que en las redes convencionales está limitado por el uso de muchos *firmware* específicos de proveedores.
- Mediante el uso de un controlador, las estadísticas pueden supervisarse a distintos niveles con respecto a cada flujo, cada puerto y cada dispositivo, superando al mismo tiempo la visión global y las posibilidades de QoS limitadas de la RT, así como la toma de decisiones por salto (Cherine y Nagy, 2022).

### 2.9.2. Conmutador OpenFlow

Un *conmutador OpenFlow* (OF-S) se refiere a un conmutador que puede soportar el protocolo OF, y puede comunicarse con el controlador (Mohammadi y Javidan, 2018).

El OF-S se compone de dos elementos principales:

- El agente OF: se encarga de la comunicación con el controlador SDN a través de mensajes *Definidos por Software* (SD) para administrar la tabla de flujos.
- La tabla de flujos: se encarga de realizar el emparejamiento de los paquetes entrantes de acuerdo con las reglas de emparejamiento y ejecutar las acciones correspondientes basadas en las reglas de entrada y la tabla de enrutamiento (GANESAN et al., 2022). La regla contiene muchas decisiones:
  - Reenvía el paquete recibido a un puerto específico.
  - Reenvío del paquete recibido al controlador.
  - Descartar el paquete recibido.
  - Inundar el paquete recibido por todos los puertos disponibles.

- Enviar al proceso normal (Shaker y Salman, 2020).

# Materiales y Métodos

## 3.1. Materiales

En este capítulo se explica los materiales usados para llevar a cabo los objetivos específicos de ésta tesis de grado, tanto para la caracterización del tráfico del escenario real, como para el diseño del escenario simulado y las mediciones de las métricas. Además se detalla los métodos utilizados para lograr el objetivo general.

En la figura 3.1 se muestra un esquema de la metodología empleada en este trabajo de grado y las diferentes etapas que se realizaron para cumplir con todos los objetivos específicos propuestos.


Figura 3.1: Esquema de la metodología empleada en esta trabajo de grado

Entre los objetivos específicos a cumplir, se tiene la caracterización del tráfico de la red de la casa de salud, caso de estudio (**clínica**), como se explicó en la sección 1.3 es posible tener un mejora en la QoS mediante el control de ciertas métricas.

Para la caracterización del tráfico de la red de la casa de salud se utilizaron:

- 2 Computadoras Portátiles.
  - Computadora **ASUS** (**pc1**) con *Windows 10*, cuenta con *Iperf3* y *Wireshark*.
  - Computadora **HP** (**pc2**) con *Windows 10*, cuenta con *Iperf3* y *Wireshark*.
- 1 Cable de red **UTP** (*Cat5e*).

Para el diseño del escenario simulado se utilizaron:

- Computadora **HP Z820 Workstation** con *Windows 10 Pro*, procesador *Intel Xeon* de 2 GHz, 16 GB de memoria RAM.
- 2 Máquinas Virtuales (*Oracle VirtualBox*).
  - Máquina virtual para el simulador MiniNet (**pcv-1**): **Ubuntu Server 22.04**, 10958 MB de memoria RAM, 7 prcesadores.
  - Máquina virtual para el controlador OpenDaylight (**pcv-2**): **Ubuntu Server 22.04**, 5000 MB de memoria RAM, 7 prcesadores.

- Iperf3.
- Wireshark.
- Simulador de red MiniNet.
- Controlador SDN OpenDaylight.

### 3.1.1. MiniNet

MiniNet es una plataforma que ofrece soporte para diversas actividades, como investigación, desarrollo, aprendizaje, prototipado rápido, pruebas y depuración. Proporciona una red experimental completa que puede ejecutarse en una computadora portátil u otra PC, permitiendo llevar a cabo estas tareas de manera conveniente y eficiente. Es una herramienta de simulación que establece una red compuesta por *hosts* virtuales, *switches*, *routers*, controladores y enlaces interconectados todo dentro de un único núcleo de Linux. Los *hosts* de MiniNet ejecutan software de red Linux convencional, mientras que sus conmutadores ofrecen soporte para OF, lo que permite un enrutamiento personalizado extremadamente flexible y la creación de redes definidas por software.

Utiliza una forma de virtualización ligera para simular una red completa en un solo sistema, compartiendo el mismo núcleo, sistema y código de usuario. Un *host* de MiniNet se comporta como una máquina real, es posible acceder a él (si se inicia el servicio *SSHD* y se conecta a la red del *host*). Estos programas pueden enviar paquetes a través de una interfaz Ethernet virtual que simula una conexión real, con una velocidad de enlace y un retardo determinados. Los paquetes son procesados por componentes virtuales que simulan *switches*, *routers* o dispositivos de red intermedios con una cantidad específica de colas. Cuando dos programas, como un cliente y un servidor *Iperf*, se comunican a través de MiniNet, el rendimiento medido debería ser comparable al de dos máquinas nativas (aunque ligeramente más lento).

Es posible crear una red MiniNet que asemeje a una red de hardware o viceversa, y ejecutar el mismo código binario y aplicaciones en cualquier plataforma. De esta manera, MiniNet ofrece flexibilidad y compatibilidad, permitiendo simular redes de *hardware* con componentes virtuales y ejecutar el mismo *software* en diferentes entornos. Está desarrollado en Python y permite la interacción con *scripts* (Fariño et al., 2022).

MiniNet proporciona las siguientes funcionalidades:

- Desarrollo rápido de prototipos para protocolos de redes novedosos.

- Realización de pruebas simplificadas para topologías complejas sin tener que invertir en hardware costoso.
- Ejecución auténtica, ya que opera con código real en los núcleos de Unix y Linux.
- Ambiente de código abierto respaldado por una comunidad numerosa que aporta una amplia documentación.

Los nodos lógicos de MiniNet pueden ser interconectados para formar redes. Estos nodos, también conocidos como contenedores o espacios de nombres de red, consumen menos recursos y permiten crear redes de más de mil nodos que pueden ejecutarse en una sola computadora portátil. Un contenedor de MiniNet es un proceso o grupo de procesos que ya no tiene acceso a todas las interfaces de red nativas del sistema anfitrión. En su lugar, se les asignan interfaces virtuales de Ethernet que se conectan entre sí mediante un conmutador virtual. Establece la conexión entre un *host* y un *switch* mediante un enlace virtual de Ethernet (*veth*). Este enlace *veth* es similar a un cable que conecta dos interfaces virtuales (of South California, s.f.).

### 3.1.2. Python

Python es considerado uno de los lenguajes de programación más accesibles, tanto para entender, como para utilizar, debido a su sintaxis legible, similitud con otros lenguajes orientados a objetos y la amplia disponibilidad de bibliotecas útiles (Fariño et al., 2022).
Python, ideado por Guido van Rossum a fines de los años 80, es un lenguaje de programación de alto nivel reconocido por su sintaxis clara y su enfoque en la legibilidad del código.
En los últimos años, Python se ha consolidado como un lenguaje de referencia tanto en proyectos industriales como científicos. Su sintaxis sencilla y su capacidad para adaptarse a diferentes paradigmas de programación (principalmente la programación orientada a objetos, aunque también permite la programación funcional) han contribuido a su amplia aceptación e integración.
Existen entornos de desarrollo integrados (IDE) específicos para Python. Los tres entornos de desarrollo libres más populares son, PyDEV es un conocido complemento de Python para el entorno de desarrollo Eclipse, ampliamente utilizado en el ámbito académico, PyCharm y Wing (Barraqué y Rodó, s.f.).
El intérprete de Python y la amplia biblioteca estándar están disponibles de manera gratuita, ya sea en forma de código fuente o binario, para todas las plataformas principales a través del sitio web de Python, https://www.python.org/. Estos pueden ser distribuidos sin restricciones.

El mismo sitio web también ofrece distribuciones y enlaces a numerosos módulos, programas y herramientas gratuitas de Python de terceros, así como documentación adicional.

Es sencillo ampliar las capacidades del intérprete de Python mediante la adición de nuevas funciones y tipos de datos implementados en C o C++ (u otros lenguajes que sean llamables desde C). Python también resulta adecuado como lenguaje de extensión para aplicaciones personalizables (van Rossum, 2018).

### 3.1.3. Iperf3

Iperf3 es una herramienta diseñada para realizar mediciones activas del ancho de banda máximo en redes IP. Permite ajustar parámetros relacionados con la sincronización, los búferes y los protocolos (TCP, UDP, SCTP con soporte para IPv4 e IPv6). Cada prueba genera informes que incluyen datos sobre el ancho de banda, pérdida de paquetes y otros parámetros relevantes. A diferencia de la versión original, iPerf3 es una nueva implementación desarrollada por ESnet/Lawrence Berkeley National Laboratory y no es compatible con versiones anteriores. Fue creado bajo la licencia de tres cláusulas BSD y no comparte código con su predecesor, iPerf, que fue desarrollado por NLANR/DAST.

Entre las carasterísticas tiene:

- TCP y SCTP
    - Medir el ancho de banda.
    - Informar el tamaño de MSS/MTU y los tamaños de lectura observados.
    - Soporte para el tamaño de la ventana TCP a través de los buffers de sockets.
- UDP
    - El cliente puede crear flujos UDP con un ancho de banda especificado.
    - Medir la pérdida de paquetes.
    - Medir la fluctuación del retardo (jitter).
    - Capacidad de multidifusión.
- Multiplataforma: Windows, Linux, Android, MacOS X, FreeBSD, OpenBSD, NetBSD, VxWorks, Solaris, etc.

Existen otras herramientas como *nuttcp*, *netperf*, *ttcp*. Pero para este trabajo de grado se decidió usar ésta herramienta ya que permite una gran flexibilidad de pruebas, porque ofrece una

variedad de opciones y configuraciones que permiten realizar pruebas de rendimiento según necesidades específicas. Permite medir tanto el rendimiento de la red TCP como UDP, ajustar parámetros de prueba como el tamaño de los paquetes, la duración de la prueba, el ancho de banda, etc. También ofrece informes detallados sobre el rendimiento de la red, incluyendo mediciones de ancho de banda, latencia, *jitter* y pérdida de paquetes. Esto te permite obtener una visión completa de la calidad y el rendimiento de la red. Además del uso eficiente de recursos, ya que está diseñado para minimizar el impacto en los recursos del sistema durante las pruebas de rendimiento. Esto significa que se puede ejecutar pruebas intensivas sin que se vea afectado el rendimiento general del sistema.

### 3.1.4. OpenDaylight

OpenDaylight (ODL) es una plataforma modular de código abierto respaldado por IBM, Cisco, Juniper, VMWare y otros importantes proveedores de redes. Es una plataforma de controlador SDN implementada en Java. Puede ser desplegada en cualquier hardware y sistema operativo que admita Java. Ofrece control centralizado y programable, así como monitoreo de dispositivos de red. Este controlador es un proyecto colaborativo organizado por la Fundación Linux y tiene como objetivo acelerar la adopción de SDN a nivel mundial (Khattak et al., 2015).

El controlador SDN de ODL se estructura en diferentes niveles. La capa superior incluye aplicaciones de lógica empresarial y de red. La capa intermedia corresponde al *framework* y la capa inferior está compuesta por dispositivos físicos y virtuales.

La capa intermedia del controlador SDN de ODL es el *framework* en el cual se manifiestan las abstracciones del SDN. Esta capa alberga las API tanto hacia el norte como hacia el sur. El controlador expone API abiertas hacia el norte que son utilizadas por las aplicaciones. ODL es compatible con el framework OSGi y REST bidireccional para la API hacia el norte. La lógica empresarial reside en las aplicaciones ubicadas por encima de la capa intermedia. Estas aplicaciones utilizan el controlador para obtener inteligencia de red, ejecutar algoritmos para realizar análisis y luego utilizar el controlador para orquestar nuevas reglas, si es necesario, en toda la red (Khattak et al., 2015).

La plataforma OpenDaylight es modular y compatible con múltiples protocolos, lo que permite a los usuarios construir un controlador SDN que se adapte a sus necesidades específicas. Esta aproximación modular y multiprotocolo brinda a los administradores de TI la flexibilidad de elegir un solo protocolo o seleccionar varios protocolos para abordar problemas complejos a medida que evolucionan. Algunos de los protocolos compatibles incluyen OpenFlow, *Open vSwitch Database* (OVSDB), *Network Configuration Protocol* (NETCONF) y *Border Gateway*

*Protocol* (BGP).

El controlador ODL se implementa exclusivamente en software y se ejecuta en su propia máquina virtual Java (JVM). Esto permite su despliegue en hardware y en sistemas operativos compatibles con Java. El controlador utiliza herramientas como Maven para la automatización de la construcción, OSGi para la carga y gestión dinámica de paquetes JAR, interfaces Java para la escucha de eventos y especificaciones, y APIs REST para servir entidades como el administrador de topología, el rastreador de hosts, el planificador de flujos y el enrutamiento estático (Eftimie y Borcoci, 2020).

Existen otros controladores SDN como ONOS (*Open Network Operating System*), Ryu, Floodlight, Faucet, Contrail Controller, pero se utilizó ODL por su amplia comunidad y soporte: ODL cuenta con una gran comunidad de usuarios y desarrolladores activos, existe una gran cantidad de tutoriales, ejemplos y casos de uso de ODL, así como contribuciones y actualizaciones frecuentes. Además permite la incorporación de diversos módulos y componentes según las necesidades específicas, lo que permite adaptar y personalizar el controlador según los requisitos del entorno y las aplicaciones específicas que se desea implementar. Permite múltiples protocolos y tecnologías.

### 3.1.5. Wireshark

Wireshark, conocido anteriormente como Ethereal, es una herramienta de análisis de redes de código abierto más utilizado a nivel mundial. Desde su inicio en 1998 por Gerald Combs, más de 500 autores han contribuido con su código para ampliar las capacidades de inspección de paquetes y protocolos de Wireshark. Ofrece una amplia y potente gama de funciones y es compatible con la mayoría de las plataformas informáticas, incluyendo Windows, OS X, Linux y UNIX. Wireshark está disponible de forma gratuita como software de código abierto y se distribuye bajo la Licencia Pública General de GNU versión 2. Para la captura en tiempo real de paquetes, Wireshark utiliza la biblioteca *pcap* (libpcap). Sus funciones de análisis incluyen el reensamblado de flujos, la decodificación y desglose de cientos de protocolos, así como la recopilación de estadísticas de tráfico y protocolo.

Cuando se utiliza para el análisis de tráfico en tiempo real, Wireshark tiene una limitación: debe ejecutarse en la misma máquina que realiza la captura de paquetes a través de la interfaz libpcap para suministrar los datos (Munz y Carle, 2008).

Wireshark cuenta con herramientas para capturar, visualizar y analizar paquetes de datos. Además, ofrece un soporte avanzado para el análisis de protocolos inalámbricos, lo que ayuda a los administradores a solucionar problemas en redes Wi-Fi. Con el soporte adecuado de los

controladores, Wireshark puede capturar tráfico “del aire” y decodificarlo en un formato que facilita la identificación de problemas que afectan el rendimiento, la conectividad intermitente y otros problemas comunes en las redes (Banerjee et al., 2010).

## 3.2. Métodos

En esta sección se mostrarán los métodos utilizados para cumplir con el objetivo general, detallando como se obtuvo las métricas óptimas empleadas en éste trabajo, y se detalla como se llevaron a cabo los experimentos para la caracterización del tráfico de red de la clínica.

### 3.2.1. Identificación de las métricas de QoS

Para evaluar las métricas de QoS, se pretende primeramente determinar las métricas de QoS para escenarios de telediagnósitico el cuál es uno de los objetivos específico planteado en este trabajo de grado. Para ello, se realizó un análisis bibliográfico con artículos **SCOPUS**, utilizando la cádena de búsqueda: *SDN AND QoS AND (image OR video) OR traffic engineering OR DICOM OR picture* con un filtro por año desde el 2018 hasta el 2022, dando cómo resultado 110 artículos. Luego de la primera revisión se tomaron 75 artículos (como se muestra en la tabla A.1), y por último se leyeron de manera completa 30 artículos para determinar las métricas óptimas (como se muestra en la tabla A.1).
En manera de resumen en la tabla 3.1 muestra el número de artículos que utilizaron diferentes métricas de QoS, se seleccionaron las métricas más empleadas según la tabla tanto, para caracterizar el tráfico del escenario real, como para el escenario simulado, las cuáles son las siguientes:

- Pérdida de paquetes.
- *Throughput.*
- *Jitter.*
- Latencia.
- Ancho de banda.

En la tabla 3.1 4 artículos han aplicado *link delay*, para el trabajo de grado no se utilizó ya que el *average end-to-end delay* proporciona una visión más completa del tiempo total de transmisión. Sin embargo, la latencia se considera como una medida del retraso o demora experimentada

en la comunicación de extremo a extremo en una red. Por lo tanto, la latencia se relaciona directamente con el *average end-to-end delay*, por ende se midió la latencia para éste trabajo de grado. Las métricas de *Peak Signal-to-Noise Ratio* (PSNR) y *Structural Simalirity Index* (SSIM), se utilizan para evaluar la calidad de imagen comprimida o modificada en comparación con la versión original, dentro de este trabajo de grado no se utilizarán imágenes médicas por ciertas limitaciones explicadas en el capítulo 3.2.2, por lo tanto, no se evaluaron éstas métricas. Como se explicó anteriormente existen dos tipos de evaluación de la QoS, cualitativa y cuantitativamente, se realizó una evaluación cualitativa de las métricas antes expuestas, por lo tanto, no se aplicó el *Mean Opinion Score* (MOS).

Seguidamente, la caracterización del tráfico del escenario real se llevó a cabo mediante la emulación de un escenario de telediagnóstico dentro de la red de la clínica, para que el escenario fuese lo más real posible, una de las computadoras se conectó a la red cableada simulando el equipo médico (tomógrafo, ecógrafo, etc.), y la otra computadora se conectó a la red inalámbrica el cuál haría de función como el computador personal del doctor como se muestra en la figura 3.2. Previamente se generaron archivos con el tamaño de distintos tipos de estudios médicos, estos fueron enviados con la herramienta Iperf3 y se capturaron los resultados con la misma herramienta y el analizador de red Wireshark, para así obtener la información necesaria

| **Métrica de QoS** | **Número de artículos** |
|---|---|
| Packet loss ratio | 16 |
| Avarage end-to-end delay | 14 |
| Throughput | 14 |
| Jitter | 14 |
| Latency | 13 |
| Bandwidth | 4 |
| Link delay | 4 |
| PSNR | 4 |
| SSIM | 3 |
| MOS | 2 |
| Path Calculation Time | 1 |
| Recovery time | 1 |
| Queue length | 1 |
| Recovery loss | 1 |
| Convergence Time | 1 |
| RTT | 1 |
| Avarage response time | 1 |
| Packet Delivery Ratio | 1 |
| Control Overhead | 1 |

Cuadro 3.1: Cantidad de veces que las métricas fueron utilizadas según el análisis bibliográfico

para determinar y evaluar las métricas de éste trabajo de grado. En la figura 3.3 se muestra un esquema dónde se muestra los pasos realizados para lograr la caracterización del tráfico desde el levantamiento de la topología hasta la captura de los diferentes tráficos.

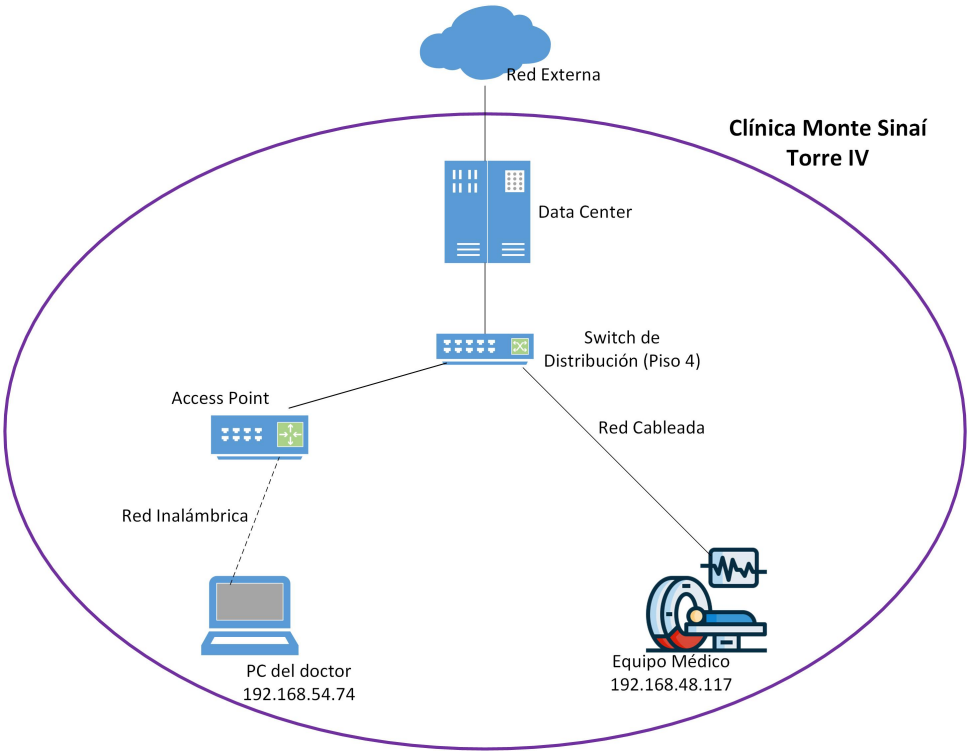



Figura 3.2: Escenario real

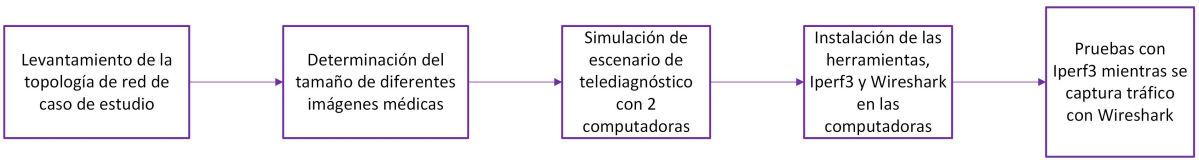



Figura 3.3: Esquema para la caracterización del tráfico del escenario real

### 3.2.2. Caracterización del tráfico de red de la Clínica Monte Sinaí

En ésta sección se define la topología de red de la clínica Monte Sinaí y el escenario de telemedicina real, también se explica cómo se realizaron las mediciones para las diferentes imágenes médicas.

La Clínica Monte Sinaí está dividida en 4 edificaciones (torres), en cada torre se concentran servicios como: consulta externa, farmacia, laboratorio. Entre las torres, la torre IV es la que cuenta con una infraestructura de red más actual, es por eso que se decidió utilizar como caso de estudio.

La torre IV consta de 5 pisos y 4 subsuelos, en la planta baja se tiene farmacia, laboratorio,

consultorios y un punto de recepción, éstos dos últimos, se tienen en todos los pisos. Cada piso cuenta con conexiones de punto de red para *Access Point* (AP), televisores, cámaras de seguridad, parlantes, y puntos de red en pared para voz/datos. En la tabla 3.2 se muestra un resumen del número de puntos de red que se encuentran habilitados en el edificio en cuestión.

| Equipo | Cantidad |
|---|---|
| Punto de red en pared para voz/datos | 185 |
| Access Point | 14 |
| Cámaras de seguridad | 38 |
| TV | 9 |
| **Total** | 246 |

Cuadro 3.2: Número de conexiones de red

En la figura **??** se muestra la topología de red de la clínica.

**Nota:** existen dos versiones de éste trabajo de grado una pública y una privada, para mantener la integridad y la seguridad de la red del caso de estudio. Esta versión corresponde a la pública se solicita total discreción, no compartir ni divulgar éstos datos a terceros, ni utilizarlos para ningún tipo de referencia.

La clínica no cuenta para la fecha con ningún escenario, implementación o aplicación de telemedicina, por lo tanto, para éste trabajo de grado se emulará un escenario de telediagnóstico, específicamente, lo que implica una atención médica síncrona. Como ya se definió anteriormente, el telediagnóstico no es más que la transmisión de imágenes médicas, historial clínico, videoconferencia, etc. al médico o a un grupo de médicos que se encuentra(n) ubicado(s) en un lugar distinto al paciente, para efectos de el enfoque de éste trabajo de grado, se simulará que un equipo médico (tomógrafo, ecógrafo, etc.) se encuentra conectado a la red cableada en un consultorio y transmita las imágenes médicas (archivos *randoms*), para que el médico las visualice en su computador portátil, el cual está conectada a la red inalámbrica como se mostró en la figura 3.2, así de ésta manera se define el escenario completo empleado en éste trabajo de grado, tanto para el escenario real, como para el escenario simulado.

La caracterización del tráfico de red se refiere al proceso de comprender y describir las propiedades y comportamientos del tráfico que fluye a través de una red. Una forma efectiva de llevar a cabo esta caracterización es mediante el análisis de métricas de QoS, que son indicadores cuantitativos que evalúan diversos aspectos del rendimiento y la eficiencia de una red.

Las métricas de QoS están diseñadas para medir diferentes elementos del comportamiento del tráfico en una red. En estudios como los realizados en (Yamansavascilar et al., 202), (Farizi et al., 2021), (Randhawa, 2020), (Marfull, 2009) analizaron diferentes métricas para describir y caracterizar el comportamiento de sus redes.

Al analizar estas métricas de QoS en el contexto del tráfico de red, es posible obtener una comprensión profunda de cómo se comporta el tráfico, cómo se ven afectadas las aplicaciones y cómo se pueden abordar los problemas de rendimiento.

Luego de describir la topología con la que cuenta el caso de estudio y definir el escenario es posible caracterizar su tráfico. Se debe analizar y describir las propiedades y comportamientos de los datos que fluyen, lo que implica el uso de diversas técnicas y métricas para comprender y categorizar con el objetivo de obtener información relevante. Por lo tanto, implica examinar y recopilar datos sobre diferentes aspectos como, volumen, patrones, características de QoS, etc. Estas pruebas se llevaron a cabo de manera presencial en las instalaciones de la Clínica Monte Sinaí Torre IV (ver figuras. A.5 y A.6).

Los experimentos para la caracterización del tráfico se realizaron en horario de alto tráfico según lo indicado por el departamento de Sistemas de la clínica, lo cual es un día entre semana en el periodo de 2:00pm a 6:00pm.

Según (DICOM, 2023a) el modelo estándar de comunicación de un DICOM, que incluye tanto la comunicación en red (en línea) como el intercambio de almacenamiento de medios (fuera de línea), las aplicaciones pueden usar cualquiera de los siguientes mecanismos de transporte:

- El Servicio de Mensajes DICOM y el Servicio de Capa Superior, que proporcionan independencia de un soporte específico de comunicación de red física y protocolos como TCP/IP.
- La API de Servicios Web DICOM y el Servicio HTTP, que permiten el uso de hipertexto común y protocolos asociados para el transporte de servicios DICOM.
- El Servicio Básico de Archivos DICOM, que proporciona acceso a los medios de almacenamiento independientemente de los formatos de almacenamiento de medios y estructuras de archivos específicos.
- La Comunicación en Tiempo Real DICOM, que ofrece transporte en tiempo real de metadatos DICOM basado en SMPTE y RTP.

Para el Servicio de Capa Superior (DICOM, 2023b) explica como se establece una conexión TCP. Y para tiempo real se utiliza SMPTE el cual es un conjunto de estándares que contribuye hacia un mecanismo común basado en protocolo de Internet (IP) para las industrias de medios profesionales. El conjunto de estándares SMPTE especifica el transporte, la sincronización y la descripción de flujos de esencia elementales separados sobre IP para producción, reproducción y otras aplicaciones de medios profesionales en tiempo real. El estándar SMPTE ST 291-1 asigna paquetes de datos auxiliares en paquetes de protocolo de transporte en

tiempo real (RTP) que se transportan a través del protocolo de datos de usuario/protocolo de Internet (UDP/IP) (SMPTE, 2023). Para éste trabajo de grado se utilizaron como protocolos de transporte TCPC y UDP los cuales son aceptados para imágenes DICOM según la literatura. Como la clínica no cuenta con aplicaciones de telediagnóstico, se simuló la transmisión de imágenes en formato DICOM, éstas pueden ser:

- Imágenes de radiografías.
- Imágenes de tomografía computarizada (TC o CT).
- Imágenes de resonancia magnética (RM o MRI).
- Imágenes de ultrasonidos.
- Imágenes de medicina nuclear.
- Imágenes de angiografía.
- Imágenes de mamografía.

Diferentes autores especifican alguno de los tamaños de las imágenes clínicas como se muestra en la tabla 3.3.

| **Fuente** | Experts, 2021 | Romero, s.f. | Rodríguez, 2016 | Dandu, 2008 |
|---|---|---|---|---|
| **Imagen Médica** | **Tamaño** | | | |
| Mastrografías | 160 MB o 280 MB | | | 27 MB (por imagen) |
| Ultrasonido | 5-60 MB | 0,0625 MB | 2,4-7,4 MB | 262 KB (por imagen) |
| Tomografía axial computarizada (TAC o TC) | 20 MB | 0,5 MB | 8,4-63 MB | 524 KB (por imagen) |
| Radiografía computarizada | 16 MB | 10 MB | 8-20 MB | 18 MB |
| Resonancia Magnética | 8 MB | 0,25 MB | 4,4-52 MB | 131 KB (por imagen) |
| Medicina Nuclear | 1-2 MB | 0,25 MB | 0,12-1,5 MB | |
| Digitalizador | | 10 MB | 9-20 MB | |
| Angiografía | | 1 MB | 4- 12 MB | |

Cuadro 3.3: Tamaño de diferentes imágenes médicas

Para éste trabajo de grado se simuló el envío de las siguientes imágenes médicas con los tamaños proporcionados por (Experts, 2021) ya que es la fuente más reciente:

- Resonancia Magnética.

- Tomografía Axial Computarizada (TAC o TC).
- Radiografía.
- Imágenes de medicina nuclear.
- Mastografías.
- Ultrasonidos.

Como se explicó anteriormente se utilizó *Iperf3* ya que sirve tanto en el escenario real, como en el escenario simulado, ésta herramienta permite enviar archivos, con lo cual se utilizó para simular la transmisión de las imágenes simuladas. Primeramente se crearon archivos *random* con los tamaños específicos que se muestran en la tabla 3.3 con un generador para test online, disponible en: https://pinetools.com/random-file-generator (ver fig. A.10).

Para las pruebas fue necesario generar dos tipos de archivos para la simulación de cada imagen médica que se envió, una para la transmisión a través de TCP (ver figura A.8), y otros para la transmisión a través de UDP (ver figura A.9). Ésto debido a que con la herramienta iperf3 es posible generar tráfico utilizando ambos protocolos, pero a la hora de utilizar el protocolo TCP se tienen algunas limitaciones. Entre éstas, se identificó luego de varias experimentos, que por lo general, envía entre 30-45 % del archivo y no se puede controlar el ancho de banda, por tanto, para las pruebas con TCP se crearon archivos de mayor tamaño de lo indicado en la tabla 3.3 para que los experimentos cuenten con un escenario más real.

Se utilizó la herramienta iperf3 tanto en la **pc1** como en **pc2** para crear un escenario cliente-servidor. Para ejecutar en el servidor iperf3, desde la línea de comandos se utiliza el comando mostrado en la figura 3.4, $\mathrm{iperf3.exe}$, éste por defecto utilizará el puerto **5201** para la comunicación.

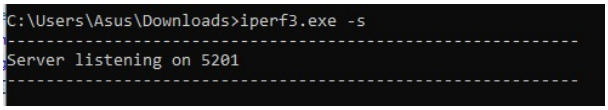



Figura 3.4: Comando para establecer el servidor

Al ejecutar la herramienta en la ventana de comandos del servidor muestra un informe que contiene los intervalos de tiempo en el que se desarrolló la prueba, el tamaño del paquete que se transfirió y el ancho de banda que se consumió (ver figura 3.5).

En el caso del cliente, como se mencionó anteriormente se realizaron pruebas con los protocolos TCP y UDP, para ésto se utilizaron los archivos antes mencionados y diferentes comandos.

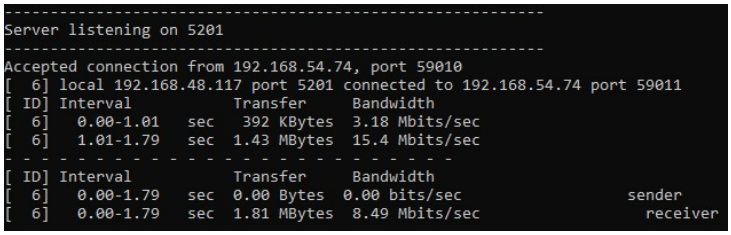



Figura 3.5: Informes de resultados desde el servidor al enviar el archivo de **nuclear** utilizando el protocolo TCP

1. **Pruebas con el protocolo TCP.** Se utilizó el siguiente comando para éstas pruebas iperf3.exe -c 192.168.48.117 -p 5201 -F [archivo de imagen médica] -t [tiempo en segundos] -V >[nombre del archivo].txt. Como se puede observar en el comando se utiliza:

   - -c: para determinar que es el cliente, seguidamente la dirección IP del servidor.
   - -p: para identificar el puerto que se desea utilizar.
   - -F: para indicar el archivo que se desea transmitir, éste se debe encontrar en la misma ubicación dónde se encuentra el iperf3.
   - -t: el tiempo que se desea realizar el testeo, debe estar en segundos.
   - -V: para mostrar más detalles dentro del informe.
   - >[nombre del archivo].txt: sirve para indicar dónde se guardará el informe, ya que de ésta manera no se mostrará en pantalla, sino que se guardará en un archivo de texto.

2. **Pruebas con el protocolo UDP.** Se utilizó el siguiente comando para éstas pruebas iperf3.exe -c 192.168.48.117 -u -p 5201 -b [ancho de banda con el que se desea transmitir][kmg o KMG] -F [archivo de imagen médica] -t [tiempo en segundos] -V >[nombre del archivo].txt. Éste comando es igual al utilizado para las pruebas TCP con la diferencia que se le agregaron dos parámetros más:

   - -u: para especificar que la transmisión se realiza a través de UDP.
   - -b: se debe especificar cuanto ancho de banda se desea utilizar para su transmisión y además se debe especificar la unidad.

En la ventana del cliente, el informe muestra información más detallada como, el protocolo de comunicación, y de la misma manera que para el servidor se señala para cada intervalo de tiempo, la cantidad de datos transferidos y el ancho de banda que se consumió. Al final del reporte indica el porcentaje que se transmitió del archivo (ver figura 3.6).

```
iperf 3.1.3
CYGWIN_NT-10.0 DESKTOP-AEBUJCB 2.5.1(0.297/5/3) 2016-04-21 22:14 x86_64
Time: Thu, 13 Jul 2023 19:43:14 GMT
Connecting to host 192.168.48.117, port 5201
      Cookie: DESKTOP-AEBUJCB.1689277394.839256.21
      TCP MSS: 0 (default)
[  4] local 192.168.54.74 port 58956 connected to 192.168.48.117 port 5201
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks, omitting 0 seconds, 200 second test
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec  3.88 MBytes  32.5 Mbits/sec
[  4]   1.00-2.00   sec  4.12 MBytes  34.6 Mbits/sec
[  4]   2.00-3.00   sec  6.00 MBytes  50.3 Mbits/sec
[  4]   3.00-3.50   sec  2.38 MBytes  40.0 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-3.50   sec  16.4 MBytes  39.3 Mbits/sec                  sender
      Sent 16.4 MByte / 38.0 MByte (43%) of radiografia1
[  4]   0.00-3.50   sec  16.4 MBytes  39.3 Mbits/sec                  receiver
CPU Utilization: local/sender 9.0% (2.6%u/6.4%s), remote/receiver 1.3% (0.3%u/0.9%s)

iperf Done.
```

Figura 3.6: Informes de resultados desde el cliente utilizando el protocolo TCP

Al realizar los experimentos con UDP en la ventana del servidor se tiene lo mismo que la figura 3.5 y en la ventana del cliente se obtuvieron informes como el que se muestra en la figura 3.7, la única diferencia con respecto al reporte proporcionado para el cliente al transmitir mediante TCP, es que éste indica el total de datagramas que envía por intervalo de tiempo y en el *summary* se muestra el *jitter* que se obtuvo en el experimento y la pérdida de datagramas.

```
iperf 3.1.3
CYGWIN_NT-10.0 DESKTOP-AEBUJCB 2.5.1(0.297/5/3) 2016-04-21 22:14 x86_64
Time: Fri, 07 Jul 2023 23:02:07 GMT
Connecting to host 192.168.48.117, port 5201
      Cookie: DESKTOP-AEBUJCB.1688770927.158341.0d
[  4] local 192.168.54.74 port 60319 connected to 192.168.48.117 port 5201
Starting Test: protocol: UDP, 1 streams, 8192 byte blocks, omitting 0 seconds, 3000 second test
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-5.00   sec  56.2 MBytes  94.3 Mbits/sec  7193
[  4]   5.00-10.00  sec  57.1 MBytes  95.8 Mbits/sec  7311
[  4]  10.00-15.00  sec  57.1 MBytes  95.8 Mbits/sec  7311
[  4]  15.00-20.00  sec  57.1 MBytes  95.8 Mbits/sec  7311
[  4]  20.00-24.59  sec  52.5 MBytes  95.8 Mbits/sec  6714
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  4]   0.00-24.59  sec   280 MBytes  95.5 Mbits/sec  1.007 ms  2486/35838 (6.9%)
[  4] Sent 35838 datagrams
      Sent  280 MByte /  280 MByte (100%) of mastografia_280MB
CPU Utilization: local/sender 26.3% (6.8%u/19.5%s), remote/receiver 5.5% (1.0%u/4.5%s)

iperf Done.
```

Figura 3.7: Informes de resultados desde el cliente utilizando el protocolo UDP

Al momento de realizar los distintos experimentos no solo se empleó iperf3 también se capturó el tráfico tanto en el cliente como en el servidor mediante Wireshark (ver figura 3.8).

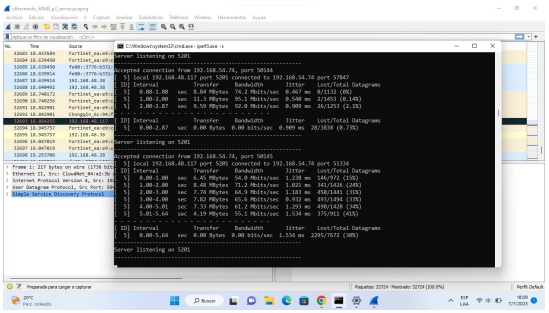


Figura 3.8: Captura de paquetes mediante wireshark en los experimentos

Los resultados de todas las métricas serán expresadas y analizadas en el capítulo 5, sin embargo, en la tabla 3.4 se muestra un resumen de los diferentes comportamientos que se obtuvo con respecto al ancho de banda. Como se explicó anteriormente al realizar experimentos con TCP no se puede controlar el ancho de banda con lo cual se quiere transmitir el archivo, sin embargo, se puede ver que las imágenes con mayor peso son las que más consumieron an-

cho de banda. En los experimentos con UDP, se logró gestionar esta métrica, lo cual condujo a la determinación de emplear el ancho de banda máximo disponible. Esta elección permitió explorar la variabilidad de la red, ya que al transmitir archivos de gran tamaño, como en el caso de las mastografías, el envío simultáneo de archivos de dimensiones similares ocasiona un uso considerable del ancho de banda, llegando incluso a generar congestión en la red.

| Protocolo de Comunicación | Imagen médica | Ancho de Banda [Mbits/sec] |
|---|---|---|
| TCP | Mastografía | 53.1 |
| | Nuclear | 8.94 |
| | Radiografía | 39.3 |
| | Resonancia | 9.62 |
| | TAC | 52.3 |
| | Ultrasonido | 54.5 |
| UDP | Mastografía [160MB] | 94.3 |
| | Nuclear [2MB] | 65.4 |
| | Resonancia | 85.8 |
| | TAC | 91.5 |
| | Ultrasonido [5MB] | 79.2 |

Cuadro 3.4: Resultados de ancho de banda para diferentes imágenes médicas, usando TCP y UDP

### 3.2.3. Definición del escenario simulado.

Para la diseño del escenario simulado se hizo uso de una computadora dónde se instalaron dos máquinas virtuales, una para desplegar el simulador MiniNet dónde se lleva a cabo la topología de red y el envío de los paquetes del tamaño de los distintos estudios de telediagnóstico y las herramientas para la obtención de las métricas como son el Iperf3 y Wireshark, y la otra máquina virtual se utilizó para desplegar el controlador SDN OpenDaylight dónde se configuraron las distintas políticas de red para este trabajo de grado. También se utilizó Python para la programación de un script el cual es la topología de red con la que cuenta el escenario real. Al obtener el diseño de la red se trata de que el escenario simulado sea lo más similar al de la clínica. Para ésto, fue necesario no solo enviar paquetes desde las dos computadoras que simulan el escenario de telediagnóstico sino también las otros dispositivos que también son parte de la arquitectura. En la figura 3.9 se muestra el esquema lógico usado para el diseño del escenario SDN.

Y por último, para la evaluación de las métricas obtenidas tanto en el escenario real, como en el escenario simulado, se realizó una comparación cuantitativa de los resultados obtenidos en ambos escenarios.

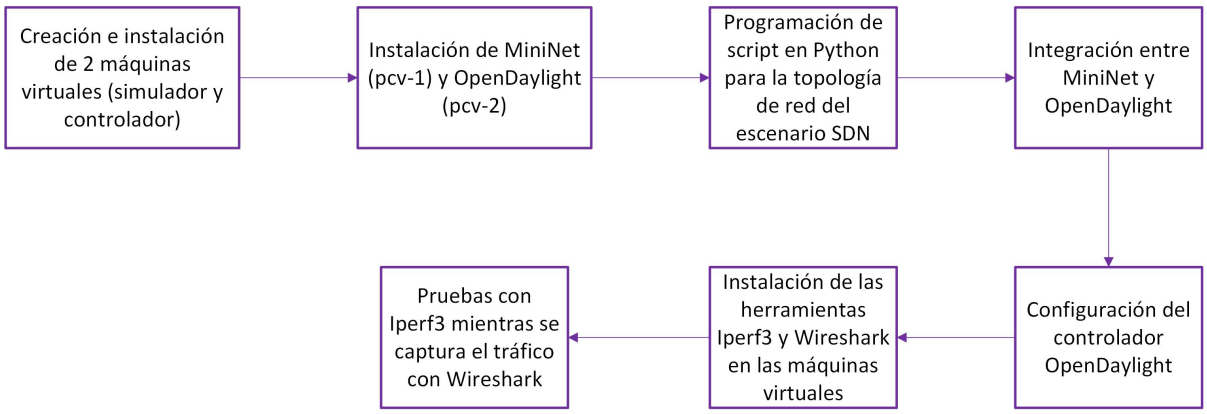



Figura 3.9: Esquema para el diseño del escenario SDN

## Diseño del escenario SDN

En este capítulo se describirá como se realizó el diseño del escenario SDN desde, la elaboración de un código en Python para la topología, el cual se utilizó para las simulaciones correspondientes a éste trabajo de grado así como, las diferentes configuraciones realizadas en el controlador de OpenDaylight, y los experimentos que se realizaron con los archivos que simulan las imágenes médicas tanto para TCP como UDP. En la figura 4.1 muestra los pasos empleados para desarrollar el escenario simulado.


Figura 4.1: Esquema para el diseño del escenario simulado SDN

Como se detalló en el capítulo 3 para la simulación se utilizaron 2 máquinas virtuales (VirtualBox), una de ellas donde se tuvo el simulador MiniNet, y la otra dónde se desplegó el controlador SDN. Las versiones utilizadas para éste trabajo de grado son: **MiniNet 2.3.1b4** (una de las versiones más actuales hasta la fecha) y **OpendayLight 0.8.0** (no se utilizó versiones superiores ya que no cuentan con la dependencia *DLUX*, la cual es la GUI del controlador y la dependencia de *VTN*). El proceso de instalación tanto de MiniNet como del OpenDaylight se encuentran detallados en el apéndice A.2 y A.3.

Luego de tener instalados ambos programas se estableció la topología de red de la clínica utilizando código en Python ($\mathrm{topologia.py}$). El código correspondiente se muestra en el apéndice A.11. Se utiliza la biblioteca $\mathrm{MiniNet}$ para crear la topología SDN simulada. El código cuenta con una serie de funciones: la función $\mathrm{cleanup\_mininet\_interfaces()}$ es utilizada para limpiar las interfaces de MiniNet antes de crear la nueva topología. Luego se tiene la función $\mathrm{myNetwork()}$: la cual es para crear y configurar la topología, primeramente, se inicializa una red $\mathrm{net}$ sin una topología específica, utilizando la dirección IP base $192.168.0.0/24$, se agrega un controlador $\mathrm{c0}$, el cual se declara como remoto, por lo tanto, se debe especificar la dirección IP y el puerto. Luego, se agregan varios *switchs* a la red, utilizando la clase $\mathrm{OVSKernelSwitch}$ para cada uno, previamente se inicializó un diccionario con una lista de nombres de *switchs* ($\mathrm{nombresSwitchs}$). Se generan *hosts* y se asignan a diferentes subredes, las cuales harán la función de las VLANs, se declararon cuantos *hosts* se tienen conectados para cada *switch*. Luego, se enlazan los *switchs* de distribución con el *switch* de acceso y se enlazan los *hosts* a sus respectivos *switchs*. Por último, se establecen opciones de enlace como el ancho de

banda.

Para la simulación se logró crear una topología similar a la existente en la clínica, sin embargo, el número de *hosts* conectados a cada uno de los *switchs* no pudo ser mayor, debido a las limitaciones computacionales que se presentaron, además tomando en cuenta la cantidad total de puntos de red habilitados que muestra la tabla 3.2 y según entrevista con la dirección de sistemas de la clínica, se indicó que los puntos de red para las TV están habilitados pero no se encuentran en uso, y considerando como criterio, que al menos la mitad del número de consultorios con los que cuenta la torre, los médicos se encuentran conectados a la red inalámbrica mediante un dispositivo, y una secretaria por piso también tiene un dispositivo conectado a la red inalámbrica se tiene un número estimado de 292 equipos conectados generando tráfico en la clínica, sin embargo, se entiende que a pesar de que la red de la clínica cuenta con un número específico de puntos de red, no se encuentran conectados ni generando tráfico todos normalmente. Por esta razón se lograron simular 136 *hosts*, lo que equivale el 46,57 % de la topología de la clínica. Dentro de la simulación se generó tráfico de *background* mediante Iperf3, y se realizaron los experimentos para obtener las métricas de QoS específicamente en dos de los *hosts* de la topología como lo indicado en la figura 3.2.

La versión del controlador utilizado para éste trabajo tiene como ventaja, que cuenta con una GUI (*Graphical User Interface*), en ella se muestran ventanas disponibles dependiendo de los componentes previamente instalados. En el apéndice A.3 se muestra dicha lista, los más utilizados fueron, Topology, Nodes y Yangman. En el primer componente es posible ver la topología de red conectada que se obtiene desde MiniNet, la segunda muestra las conexiones, y la última sirve para configurar el controlador, en ella se despliegan una serie de módulos disponibles (solo las previamente instaladas) y desde aquí se pueden configurar las diferentes *API REST*.

Usando la terminal de **pcv-1** (MiniNet) se deberán ejecutar los siguientes comandos: para la inicialización del controlador remoto y los *switchs* sudo python3 topologia.py; para la prueba de conectividad pingall el cual realizará pings entre todas los *hosts*, para cargar la topología completa en el controlador ver figura 4.2.

Usando el componente Nodes es posible ver los nodos (OF-S) y la cantidad de hosts conectados a ellos (ver fig. 4.3). Al ingresar en alguno de los *switchs*, desde la columna *node connectors* se muestra una tabla con los ID de cada conector, la interfaz, el número de puerto físico y la dirección MAC (ver figura 4.4).

Existen diferentes métodos de provisión de QoS que se pueden aplicar a las SDNs entre ellas se tienen:

- Mecanismos de aplicación y control: son métodos y técnicas utilizadas para gestionar y

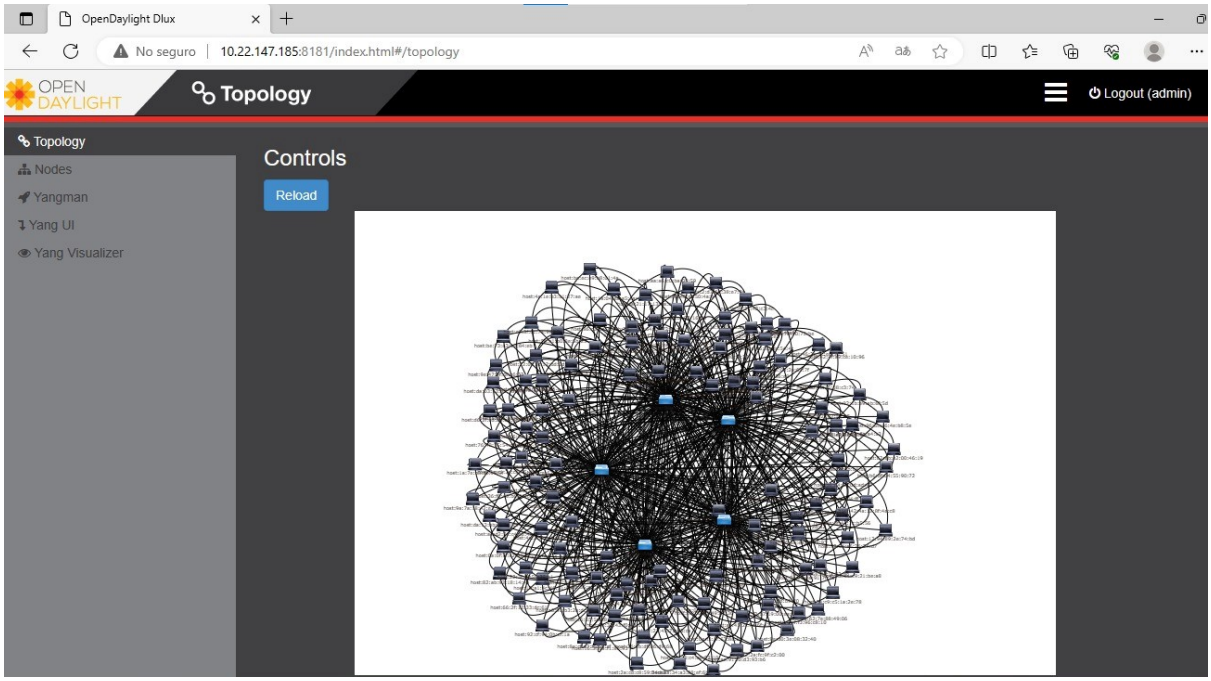



Figura 4.2: Interfaz Gráfica de OpenDaylight

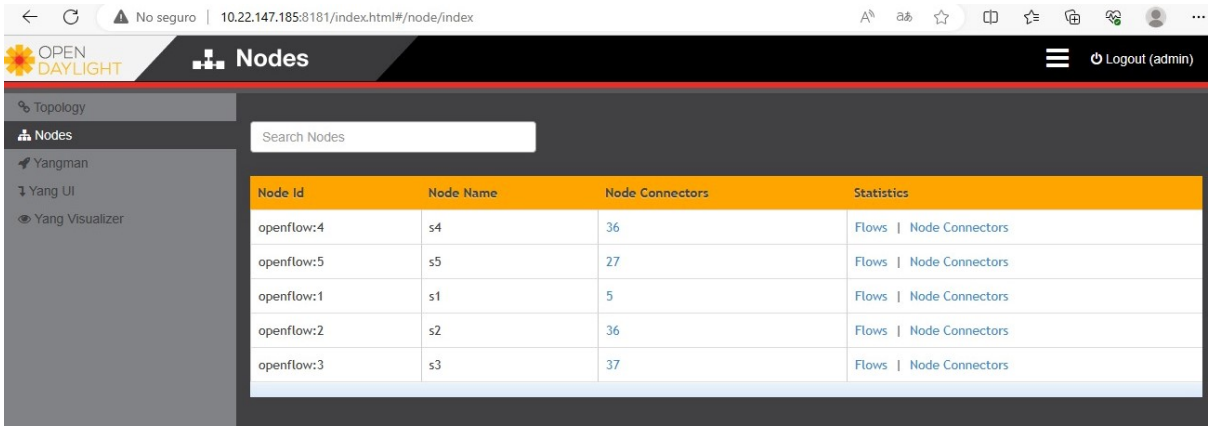



Figura 4.3: Nodos de la simulación

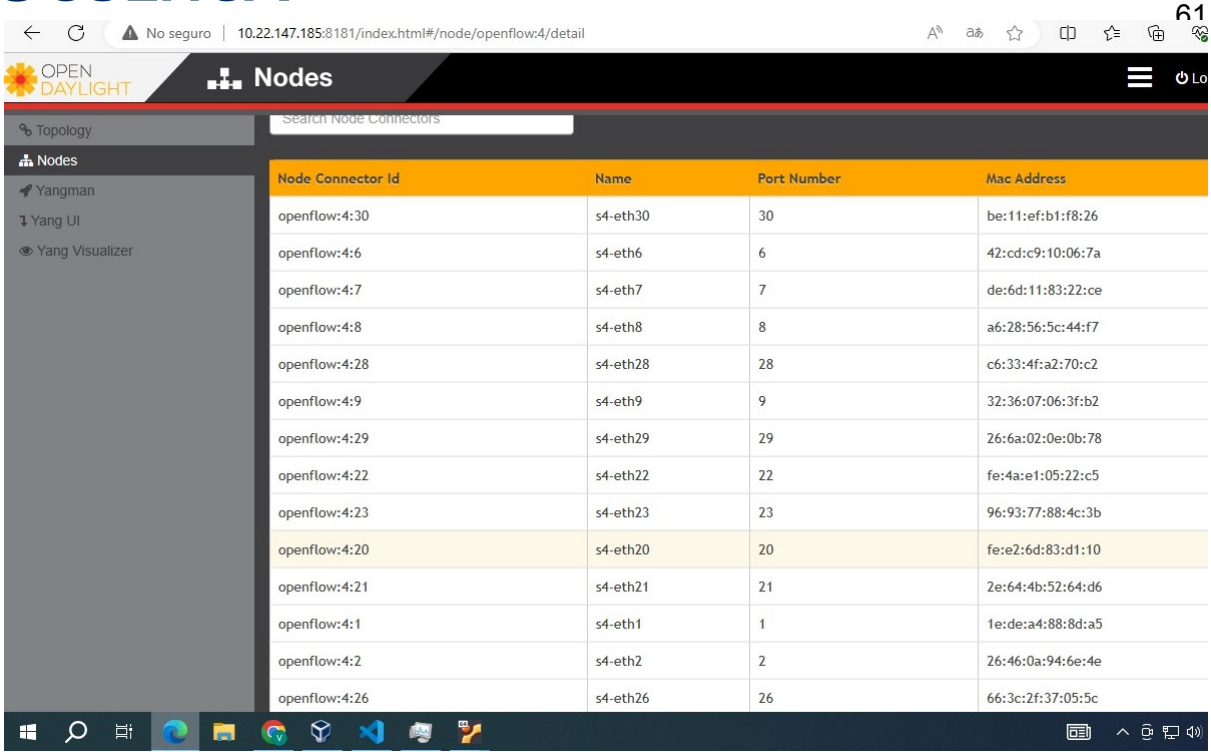



Figura 4.4: Tabla de hosts conectados a un switch

asegurar un rendimiento deseado y consistente en redes y sistemas de comunicación. Estos mecanismos se implementan para garantizar que ciertos flujos de tráfico o aplicaciones reciban el nivel adecuado de recursos y priorización en función de sus requisitos y prioridades. Para el caso de planificación existen soluciones como: (1) colas FIFO (*First-In-First-Out*), (2) colas con prioridad, (3) colas equitativas. Para el caso mecanismos de control de flujo de paquetes se tiene; (1) algoritmos de regulación por goteo, (2) regulación por testigos.

- Mecanismos de control de tráfico: son enfoques y técnicas utilizados para gestionar y controlar el flujo de tráfico en redes de comunicación. Dentro de éstos mecanismos se tiene: (1) servicios integrados (*IntServ*), (2) servicios diferenciados (*DiffSev*).

Para éste trabajo de grado se uso un conjunto de enfoques aunque principalmente están relacionados a la implementación de políticas de QoS. Dado que se abarcan aspectos como mapeo de puertos, establecimiento de condiciones de flujo, filtros para paquetes y asignación de flujos según direcciones MAC, estos se alinean más con mecanismos de control y políticas de QoS. Sin embargo, al establecer condiciones de flujo, y filtros para paquetes, esto es parte de las políticas de control de admisión y asignación de recursos en *IntServ*. También, el uso de mapeo de puertos se alinea con la clasificación y diferenciación de tráfico en categorías o clases en *DiffServ*.

Dentro del controlador se realizaron diferentes configuraciones tanto para lograr la simulación del escenario real como para utilizar algunas de las características de las tecnología SDN. Todas las API REST configuradas para este trabajo de grado se encuentran en el apéndice A.12.

Se realizaron configuraciones de **VTNs** (*Virtual Tenant Network*), ésto es una abstracción lógica de red que permite la creación y gestión de redes virtuales independientes dentro de una infraestructura física compartida. En un entorno de SDN, una VTN es similar a una red virtualizada que puede ser configurada y administrada de manera separada de otras redes, incluso si comparten la misma infraestructura física.

Las configuraciones realizadas para las VTNs son:

- **Generación de VTNs:** se generaron VTNs para cada subred con la que cuenta la clínica (ver fig. 4.5).

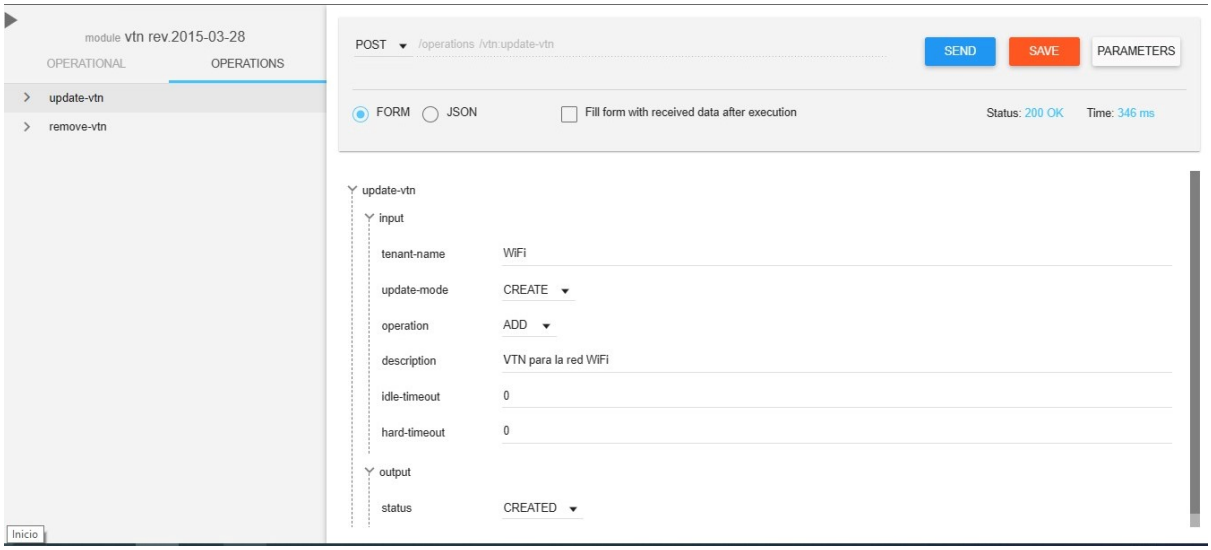



Figura 4.5: Creación de VTN

Para la generación de cualquier API REST, ODL cuenta en su GUI dos maneras de lograrlo, ya sea generando la petición JSON o con la sección de FORM el cual indica los campos que se deben llenar, además que cuenta con una ayuda dónde explica que se espera que se ingrese en cada campo.

Al llenar todos los datos que se piden como se muestra en la figura 4.5 se genera como respuesta la petición en formato JSON.

```
{
''vtn'': [
    {
```

```
            ''name'': ''SinaiFarm'',
            ''vtenant-config'': {
                ''description'': ''VTN de la red SinaiFarm'',
                ''hard-timeout'': 0,
                ''idle-timeout'': 0
            }
        }
    ]
}
```

Listing 4.1: Creación de VTN

Dentro de la creación de las VTNs se debe definir el hard-timeout y el idle-timeout, son dos parámetros que se utilizan al crear flujos en la VTN. Estos parámetros determinan la duración de un flujo y cómo debe ser tratado cuando está inactivo. Para todas las VTN se configuraron éstos parámetros con 0, lo que implica que los flujos permanecerán activos en el *switch* de red de forma indefinida.

- **Creación de los Virtual-Bridge:** se definieron los diferentes V-Bridge para cada una de las VTNs, éstos puentes actúan entre los diferentes dispositivos conectados a ellos dentro de la red virtual, lo que significa que el tráfico entre esos dispositivos se mantiene dentro de la misma red virtual y no se propaga a otras redes virtuales o a la red física subyacente. De esta manera se logra subdividir la red como la clínica sin tener que configurar VLANs (ver figura 4.6). Se crearon puentes para cada subred y para cada piso.

  Para la creación de las V-Bridge es necesario crear la VTN previamente y dentro del API REST se debe indicar el nombre del puente y es posible escribir una descripción como se muestra en el *listing* 4.2:

```
{
''update-vbridge'': {
    ''input'': {
        ''update-mode'': ''CREATE'',
        ''operation'': ''SET'',
        ''tenant-name'': ''SinaiFarm'',
        ''bridge-name'': ''sinaifarm_PB'',
```

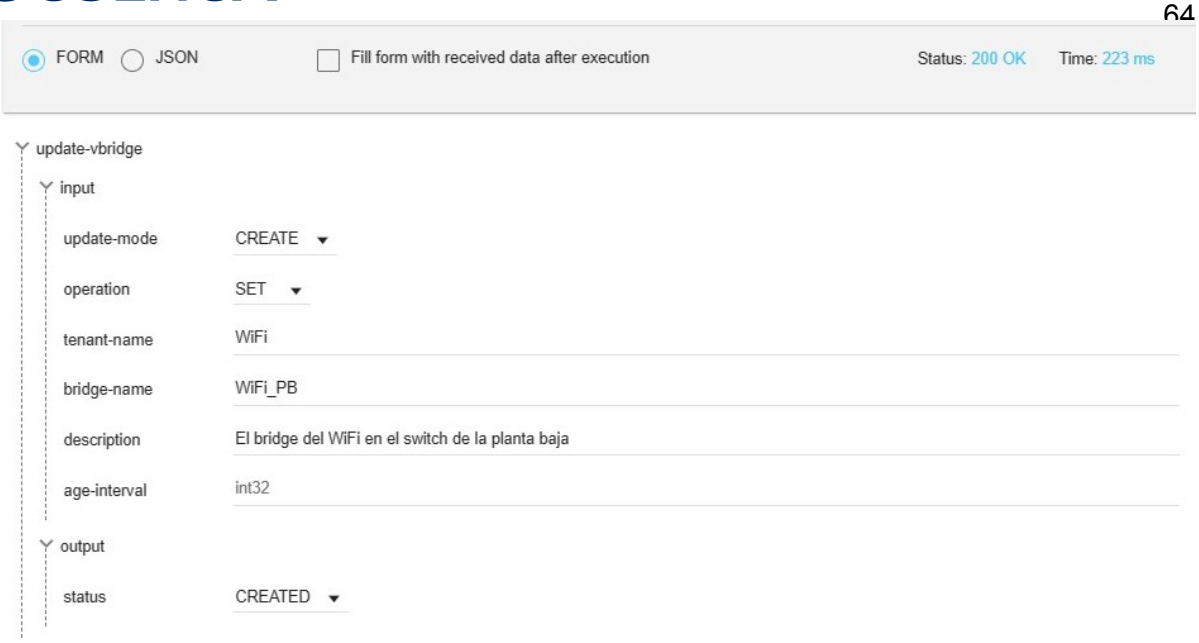



Figura 4.6: Creación de V-Bridge

```
            ''description'': ''Bridge de la red SinaiFarm en el
               switch de planta baja''
        }
    }
}
```

Listing 4.2: Creación de V-Bridge

- **Creación de las Virtual-Interface:** las V-Interface se refieren a interfaces lógicas virtuales que se crean en los nodos de red para permitir la conectividad entre los diferentes dominios virtuales. Se creo una interfaz para cada V-Bridge, en la figura 4.7 se muestra su configuración y seguidamente se muestra la solicitud en formato JSON.

```
{
''update-vinterface'': {
    ''input'': {
        ''update-mode'': ''CREATE'',
        ''operation'': ''SET'',
        ''tenant-name'': ''Medicos'',
        ''bridge-name'': ''medicos_P1'',
        ''interface-name'': ''medicosP1_1'',
        ''description'': ''Interfaz de la VTN de medicos
```

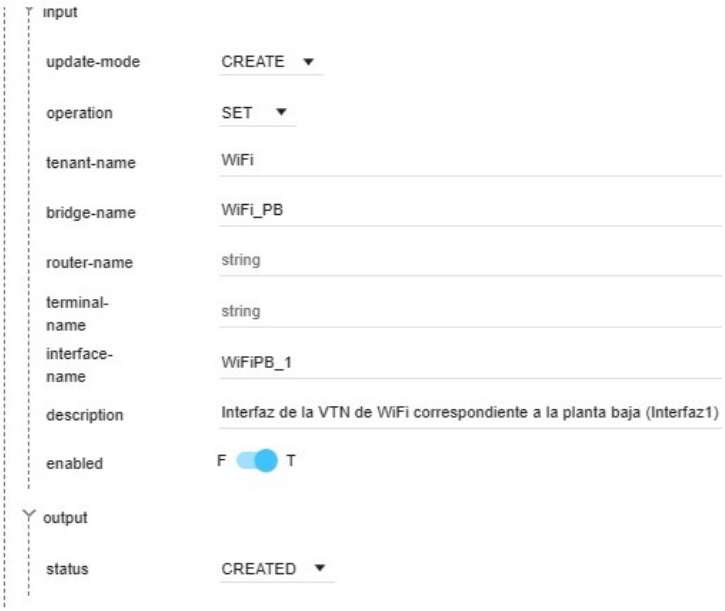



Figura 4.7: Creación de V-Interface

```
            correspondiente a la primera planta (Interfaz 1)
            '',
        ''enabled'': ''true''
    }
  }
}
```

Listing 4.3: Creación de V-Interface

- **Configuración del mapeo puertos:** se refiere a la configuración que permite conectar los puertos de un nodo (*switch* virtual o puerta de enlace) de una VTN con los puertos de un nodo físico o con otro nodo virtual en la red. Esto se hace para que los paquetes de datos puedan fluir entre las distintas redes virtuales y el resto de la red física o virtual (ver figura 4.8).

  El mapeo de puertos se muestra en la figura 4.8

```
{
''set-port-map'': {
    ''input'': {
        ''tenant-name'': ''SinaiFarm'',
        ''bridge-name'': ''sinaifarm_PB'',
        ''interface-name'': ''sinaifarmPB_1'',
```


Figura 4.8: Configuración del mapeo de puertos

```
            ''node'': ''openflow:2'',
            ''port-id'': ''4'',
            ''port-name'': ''s2-eth4'',
            ''vlan-id'': ''0''
        }
    }
}
```

Listing 4.4: Creación del mapeo de puertos

Se establecieron mapeo de puertos según las capacidades físicas del *switch* y la cantidad de conexiones necesita cada subred según la topología proporcionada por la clínica.

- **Establecimiento de condiciones de flujo:** Existen diferentes formas de establecer reglas de QoS o SLA en una red para garantizar un rendimiento adecuado. Una de ellas es gestionar el tráfico de la red mediante condiciones de flujos, lo que generará *match* en las tablas de flujo. Dentro del controlador existen 3 formas diferentes de establecer estás condiciones: (1) *vtn-ether-match* el cual se utiliza para definir las condiciones de coincidencia relacionadas con los campos de la capa de enlace (Ethernet) de un paquete; (2) *vtn-inet-match* se utiliza para definir las condiciones de coincidencia relacionadas con los campos de la cabecera IP de un paquete, como direcciones IP de origen y destino; y, (3) *vtn-layer4-match* sirve para definir condiciones de coincidencia relacionadas con los puertos de origen y destino de los paquetes TCP, UDP e ICMP en una VTN, permitiendo

aplicar reglas de flujo basadas en criterios específicos de la capa de transporte. Se utilizó está última y se configuraron para TCP y UDP (ver figura 4.9).


Figura 4.9: Establecimiento de la condición de flujo

Para el establecimiento de esta condición de flujo tanto para TCP como para UDP es necesario establecer el rango de puertos, es evidente que en la realidad este rango no debe ser tan amplio y debe estar sujeto a las posibilidades de navegación que se le quiera permitir a los usuarios. En este caso el rango es más amplio ya que las pruebas realizadas en éste trabajo son utilizando Iperf3 y a pesar de que se le indica un puerto para la conexión, éste utiliza puertos aleatorios dentro del rango establecido en ésta condición de flujo, de ésta manera se asegura que es posible realizar los experimentos. La solicitud JSON se muestra el *listing* 4.5:

```
{
''set-flow-condition'': {
    ''input'': {
        ''operation'': ''SET'',
        ''present'': ''false'',
        ''name'': ''paquetesUDP'',
        ''vtn-flow-match'': [
```

```
                {
                    ''index'': ''3'',
                    ''udp-source-range'': {
                        ''port-from'': ''1024'',
                        ''port-to'': ''49151''
                    },
                    ''udp-destination-range'': {
                        ''port-from'': ''49152'',
                        ''port-to'': ''65535''
                    }
                }
            ]
        }
    }
}
```

Listing 4.5: Establecimiento de la condición de flujo

- **Configuración del filtro del flujo:** luego de establecer todas las condiciones de flujo que se se desea es necesario indicar los filtros, es decir, en dónde se deben aplicar dichas condiciones, por tanto se debe indicar a que VTN y a cual interfaz y puente se debe establecer como se muestra en la figura 4.10 y en la salida JSON que se muestra en el *listing* 4.6.

```
{
''set-flow-filter'': {
    ''input'': {
        ''output'': ''true'',
        ''tenant-name'': ''WiFi'',
        ''bridge-name'': ''WiFi_P1'',
        ''interface-name'': ''WiFiP1_1'',
        ''vtn-flow-filter'': [
            {
                ''condition'': ''paquetesTCP'',
                ''index'': ''1'',
                ''vtn-pass-filter'': {}
```

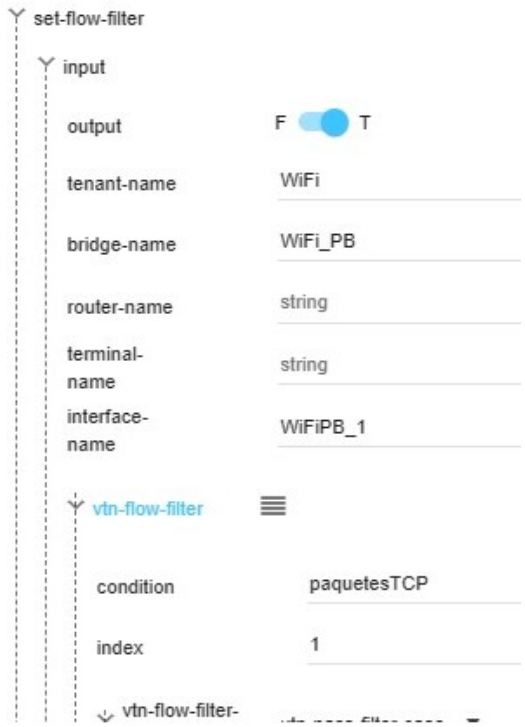



Figura 4.10: Configuracion del filtro del flujo

```
            }
          ]
        }
      }
    }
```

Listing 4.6: Configuración del filtro de flujo

- **Mapeo de MAC:** y por último se realiza un mapeo por MAC lo cual es una característica que permite definir una asignación de direcciones MAC (*Media Access Control*) específicas en una red virtual de VTN. Se utiliza para mapear direcciones MAC de hosts o dispositivos dentro de una red virtual VTN para que sean reconocidos y permitidos en la red, de esta manera se controla el acceso y la conectividad de dispositivos dentro de una red virtual, permitiendo establecer políticas de seguridad y restricciones para determinadas direcciones MAC, lo que proporciona una forma de autorizar o denegar el acceso de hosts específicos a la red virtual. Se establecieron dos mapeos por MAC correspondientes a los dos equipos que emulan el escenario de telediagnóstico como se muestra en la figura 4.11 y en la salida de JSON mostrada en el *listing* 4.7,

```
{
```

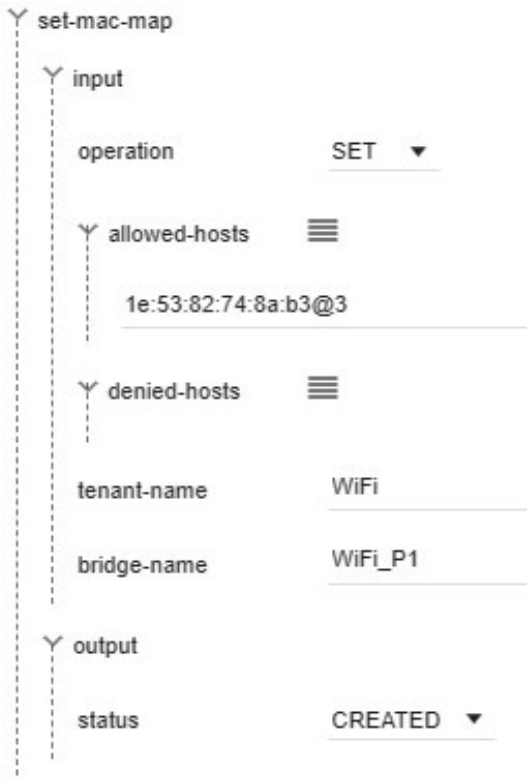



Figura 4.11: Configuracion de mapeo por MAC

```
''set-mac-map'': {
    ''input'': {
        ''operation'': ''SET'',
        ''allowed-hosts'': [
            ''da:09:00:00:5d:35@10''
        ],
        ''tenant-name'': ''Medicos'',
        ''bridge-name'': ''medicos_P3''
    }
  }
}
```

Listing 4.7: Configuración de mapeo por MAC

Luego de establecer todas las configuraciones en el controlador se realizaron los mismos experimentos descritos en la sección 3.2.2 con los mismos archivos *randoms* que simulan la transmisión de imágenes médicas. La única diferencia entre los experimentos realizados en la clínica y los realizados en la simulación es que fue necesario no solo generar tráfico con Iperf3 mediante los dos *hosts* que emulan el escenario de diagnóstico, sino que también se utilizaron la mayor cantidad de *hosts* que las máquinas virtuales permitieron para generar tráfico entre ellos, de esa manera se logró un escenario que se asemejó a la situación usual de la clínica (ver figura 4.12).

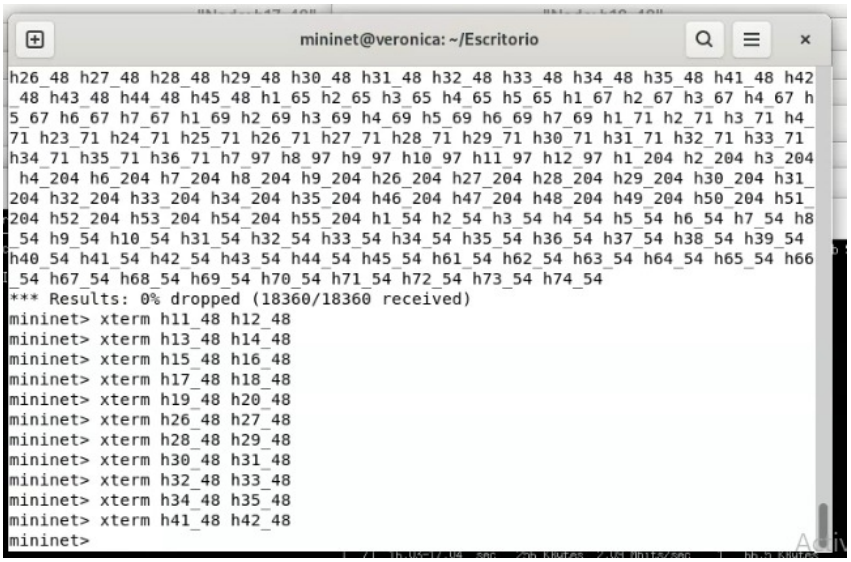



Figura 4.12: Inicialización de hosts dentro del simulador

## Resultados y Discusiones

En este capítulo se analizaron los resultados obtenidos al transmitir los archivos que emulan las imágenes médicas mediante el protocolo TCP y UDP, en el escenario real (sin SDN) y simulado (con SDN). A pesar de ser archivos generados, en éste documento se hace referencia a los archivos de imágenes médicas. Todos los resultados analizados en ésta sección para los dos escenarios muestran el comportamiento de la red influenciado por el tráfico de *background*. Además, en la sección 5.2.5 se mencionan los diferentes limitaciones que se suscitaron al realizar este trabajo de grado y las soluciones propuestas.

Se analizaron los resultados obtenidos de la siguiente manera, primero se discutieron las métricas de cada imagen mediante la transmisión con el protocolo TCP y luego mediante la transmisión con UDP. Las imágenes están ordenadas de mayor a menor tamaño en la tabla 5.1.

| Protocolo de Comunicación | Imagen médica | Tamaño de la imágen [MB] |
|---|---|---|
| TCP | Mastografía | 410 |
| | Ultrasonido | 71 |
| | TAC | 52 |
| | Radiografía | 39 |
| | Resonancia | 21 |
| | Nuclear | 7 |
| UDP | Mastografía | 160 |
| | TAC | 51 |
| | Resonancia | 16 |
| | Ultrasonido | 5 |
| | Nuclear | 1 |

Cuadro 5.1: Tamaño de los archivos que emulan las imágenes médicas utilizadas para los experimentos usando TCP y UDP

### 5.1. Análisis de métricas de QoS de imágenes médicas utilizando el protocolo TCP

Se analizarán primeramente las métricas obtenidas para el caso de los experimentos que se llevaron a cabo utilizando el protocolo de comunicación TCP, éste se caracteriza por ser un protocolo fiable, el cual es empleado para transmitir datos sensibles a la pérdida de paquetes. Sin embargo, como se explicó en la sección 3.2.2 cuando se realizaron los experimentos con Iperf3 usando TCP en la clínica, los primeros resultados obtenidos mostraban una pérdida de paquetes evidente ya que se enviaba entre el 30-45 %.

### 5.1.1. Throughput

Esta métrica se obtuvo mediante el informe adquirido por Iperf3 desde el cliente en el cual mostraba en una columna el *bitrate* en cada instante del experimento, y en otra columna se tienen los instantes de tiempo. Es una métrica importante para evaluar la eficiencia y la capacidad de un sistema para manejar datos.

- **Mastografía**

  La mastografía es el experimento con mayor tamaño de imagen, es decir, que durante éste experimento se le exigió a la red transferir un volumen grande de datos lo que usualmente no ocurre recordando que la clínica no cuenta con ningún tipo de escenario de telemedicina. Esto implica que la red es usada en gran parte para enviar datos de mucho menor tamaño como por ejemplo, para enviar correos electrónicos, enviar mensajes, uso de redes sociales, etc.

  Las gráficas que se muestran en la figura 5.1 indican el *throughput* obtenido durante el experimento (tiempo en el que se transfirió el archivo).

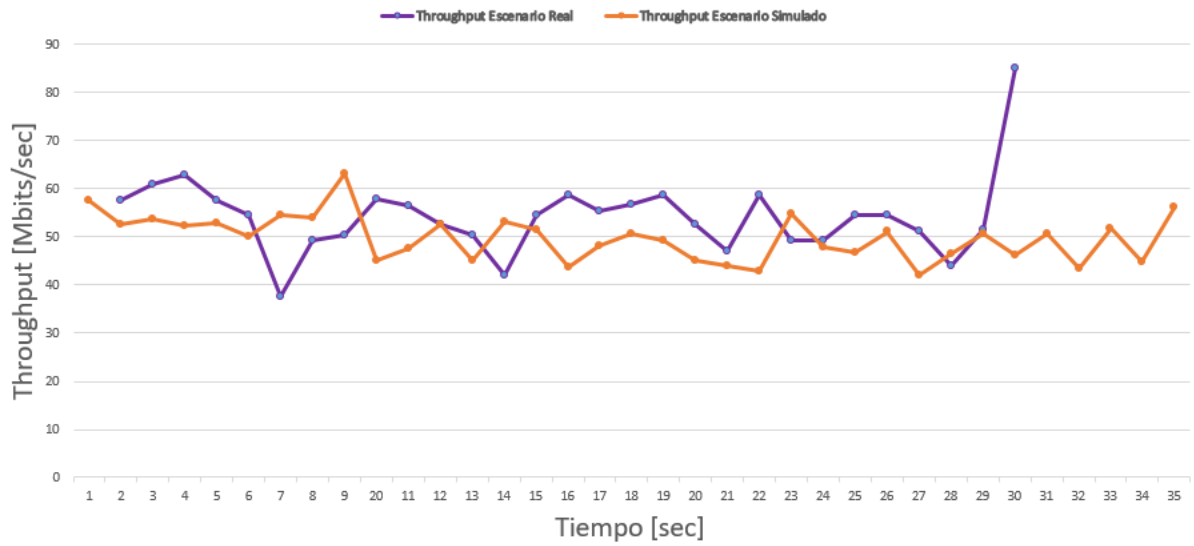



Figura 5.1: *Throughput* de mastografía transferida mediante TCP en escenario real y simulado

Para el escenario simulado los valores máximos que se obtuvieron son de 63,1 Mbps y 57,6 Mbps, en el caso del escenario real los valores máximos son, 85,1 Mbps y 62,9 Mbps. Los valores mínimos, en el caso del escenario simulado son de, 42,15 Mbps y 42,9 Mbps, para el caso del escenario real son de, 37,7 Mbps y 42 Mbps. Esto indica que se tienen picos más grandes en la clínica que en el escenario simulado.

Tanto para el escenario real como para el escenario simulado existe una estabilidad

promedio durante el experimento, con excepción del pico que muestra el escenario real en el último instante de tiempo de la transmisión, que obtuvo un *throughput* de 85 Mbps, ésto se debe a que por ser el último segmento de la transferencia, éste es mucho más pequeño en comparación a los otros segmentos enviados durante el experimento por lo cual no requiere tanto ancho de banda, lo que se traduce a un aumento en el *throughput*. Para éste experimento el *throughput* muestra una escasa mejoría en el escenario real a comparación del escenario simulado.

- **Ultrasonido**

  Las gráficas que se muestran en la figura 5.2 indican el *throughput* obtenido durante el experimento (tiempo en el que se transfirió el archivo).

  Para el escenario simulado los valores máximos que se obtuvo son de 56,3 Mbps y 51,6 Mbps, en el caso del escenario real los valores máximos son, 58,8 Mbps y 58,7 Mbps. Los valores mínimos, en el caso del escenario simulado son de, 42,26 Mbps y 43,18 Mbps, para el caso del escenario real son de, 41,9 Mbps y 50,4 Mbps. Esto indica que se tienen picos más grandes en la clínica que en el escenario simulado.

  Tanto para el escenario real como para el escenario simulado existe una estabilidad promedio durante el experimento. Se puede notar que la franja de las variaciones en el escenario real se mantiene entre 50-60 Mbps, mientras en el simulado solamente entre el 40-50 Mbps mayoritariamente, por lo tanto, el *throughput* promedio es mejor en el escenario real. En ninguno de los escenarios se tienen caídas bruscas o picos repentinos en el *throughput*.

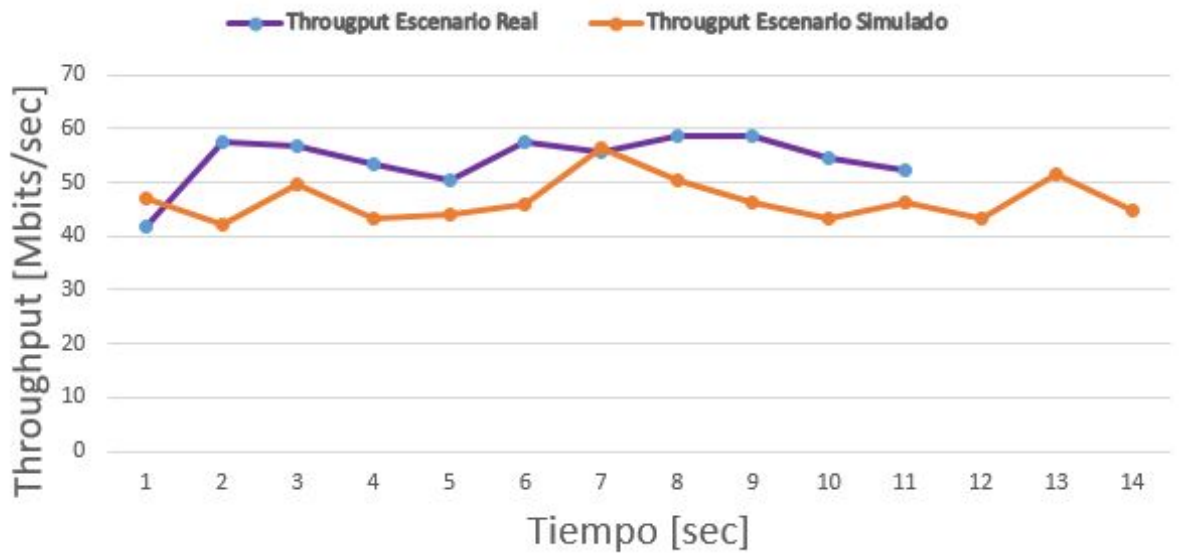



Figura 5.2: *Throughput* de ultrasonido transferido mediante TCP en escenario real y simulado

- **TAC.**

  Las gráficas que se muestran en la figura 5.3 indican el *throughput* obtenido durante el experimento. Entre los dos escenarios se muestran unas tendencias de estabilidad, en el caso del escenario simulado, el *throughput* se mantiene constante en la franja de los 42-48 Mbps, en el caso del escenario real aumenta durante el experimento pero luego disminuye en el último instante de tiempo, pero se mantiene mayor a 50 Mbps.

  Para el escenario simulado los valores máximos que se obtuvo son de 47,38 Mbps y 47,35 Mbps, en el caso del escenario real los valores máximos son, 54,5 Mbps y 53,5 Mbps. Los valores mínimos, en el caso del escenario simulado son de, 42,9 Mbps y 43,73 Mbps, para el caso del escenario real son de, 49,2 Mbps y 51,9 Mbps. Esto indica que se tienen picos más grandes en la clínica que en el escenario simulado y la clínica no muestra valles durante el experimento.

  Por lo tanto, se presenta un alto impacto en ésta métrica en el escenario simulado, ya que ni el valor más alto obtenido en las simulación, que es de 47,38 Mbps llega a ser el valor más bajo obtenido en el escenario real.

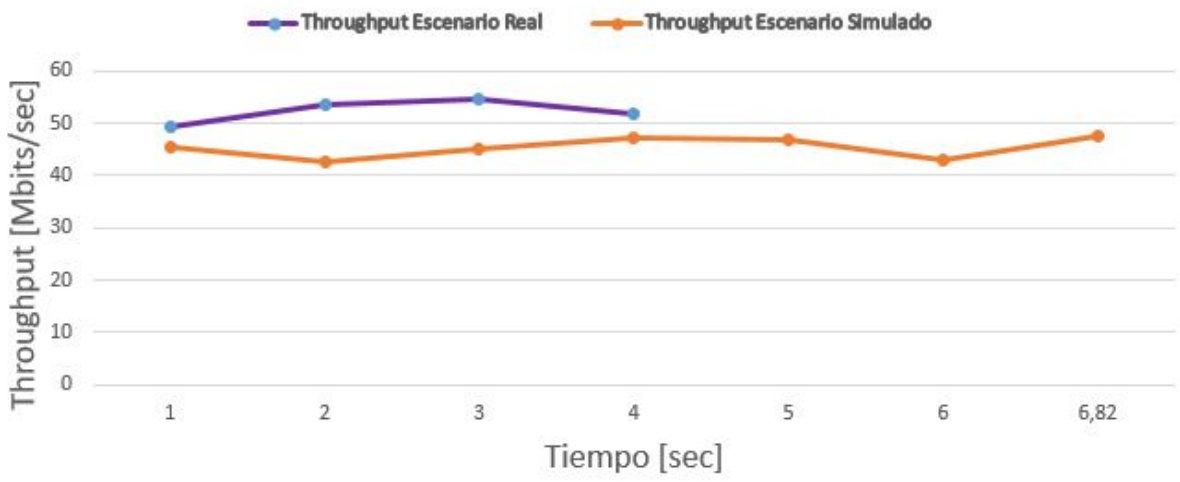



Figura 5.3: *Throughput* de TAC transferido mediante TCP en escenario real y simulado

- **Radiografía**

  Las gráficas que se muestran en la figura 5.4 indican el *throughput* obtenido durante el experimento. En ambos escenarios no se muestran diferencias significativas con respecto al comportamiento del *throughput* durante el experimento.

  Para el escenario simulado los valores máximos que se obtuvieron son de 33,1 Mbps y 31,5 Mbps, en el caso del escenario real los valores máximos son, 50,3 Mbps y 40 Mbps. Los valores mínimos, en el caso del escenario simulado son de, 24 Mbps y 24,87 Mbps, para el caso del escenario real son de, 32,5 Mbps y 34,6 Mbps. A pesar de los valores bajos de *throughput* con respecto a los obtenidos en la clínica existe una estabilidad notorea durante el experimento, no se muestran picos o caídas abruptas en el escenario simulado.

  En este experimento se puede observar como entre los instantes de tiempo del escenario simulado no existen saltos muy grandes para los valores del *throughput*, esto se debe a que el archivo por ser más pequeño en comparación a los analizados anteriormente, al segmentar el paquete se convierten en segmentos más pequeños lo cual no requiere tanto ancho de banda y es más manejable para su transmisión. Se concluye que en general se tuvo mejor *throughput* en el escenario real que en el simulado.

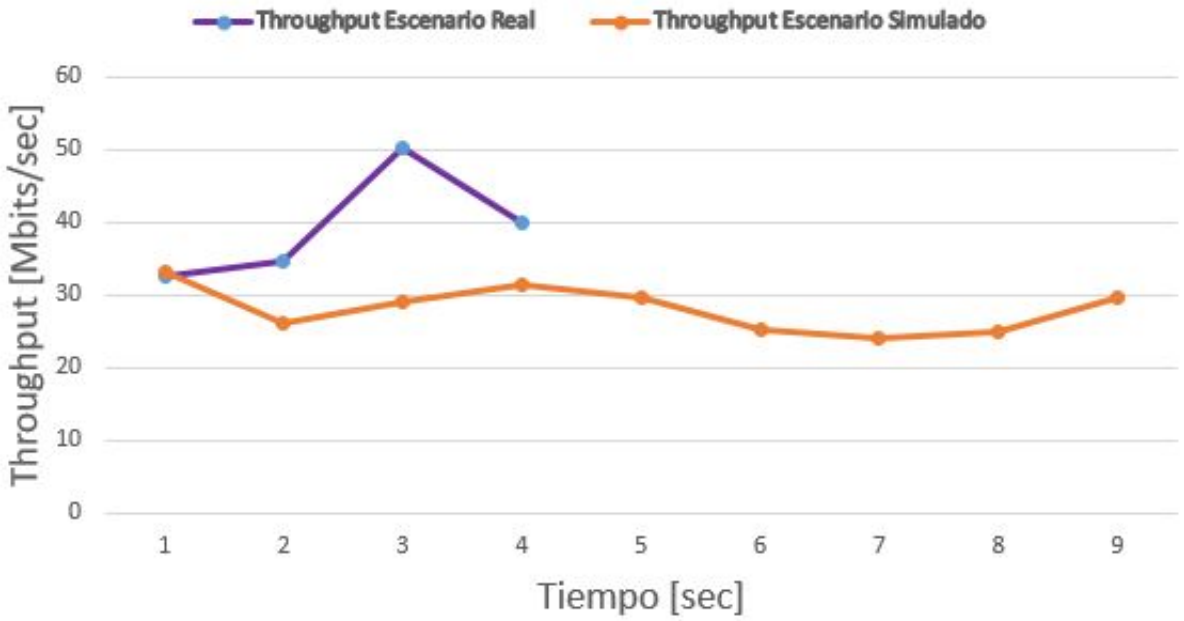



Figura 5.4: *Throughput* de radiografía transferido mediante TCP en escenario real y simulado

- **Resonancia.**

  Las gráficas que se muestran en la figura 5.5 indican el *throughput* obtenido durante el experimento. En el caso del escenario simulado se tardó más el experimento que lo que se tardó en el escenario real, por lo cual existen menos valores que analizar entre un escenario y otro.

  Para el escenario simulado los valores máximos que se obtuvieron son de 47,9 Mbps y 35,19 Mbps, en el caso del escenario real el valor máximo es de, 56,7 Mbps. Los valores mínimos, en el caso del escenario simulado son de, 31,08 Mbps y 32,5 Mbps, para el caso del escenario real es de, 48,2 Mbps, lo que implica que se tuvo mejor *throughput* en el escenario real que en el simulado. El mínimo del escenario real está sobre el valor con el que empieza el escenario simulado, por lo que se evidencia la mejora en el escenario real.

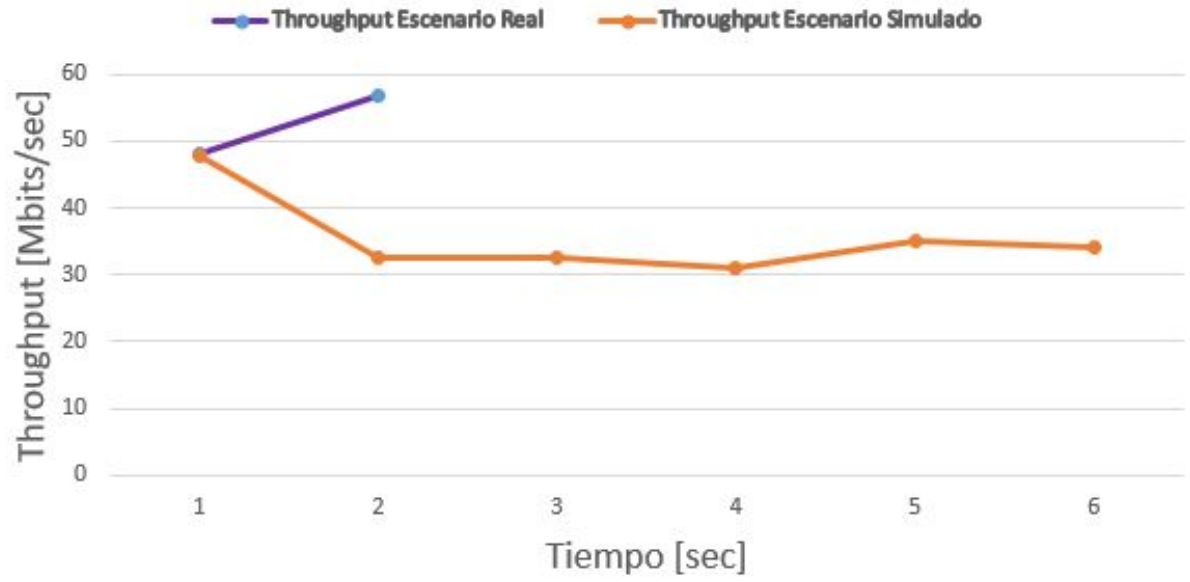



Figura 5.5: *Throughput* de resonancia transferido mediante TCP en escenario real y simulado

- **Nuclear**

  Las gráficas que se muestran en la figura 5.6 indican el *throughput* obtenido durante el experimento. En ambos escenarios se puede ver que hay una tendencia ascendente, sin tomar en cuenta la caída abrupta que se muestra en el escenario simulado en el segundo 2, lo cual se debe a factores propios de la capacidad de procesamiento de la computadora.

  Para el escenario simulado los valores máximos que se obtuvieron son de 35 Mbps y 19,1 Mbps, en el caso del escenario real el valor máximo es de, 13,8 Mbps. Los valores mínimos, en el caso del escenario simulado son de, 1,56 Mbps y 18,4 Mbps, para el caso del escenario real es de, 5,24 Mbps, lo que implica que se tuvo mejor *throughput* en el escenario simulado ya que para éste se inició con un *throughput* por encima del valor máximo del escenario real, y terminó el experimento con más del doble del valor máximo del real.

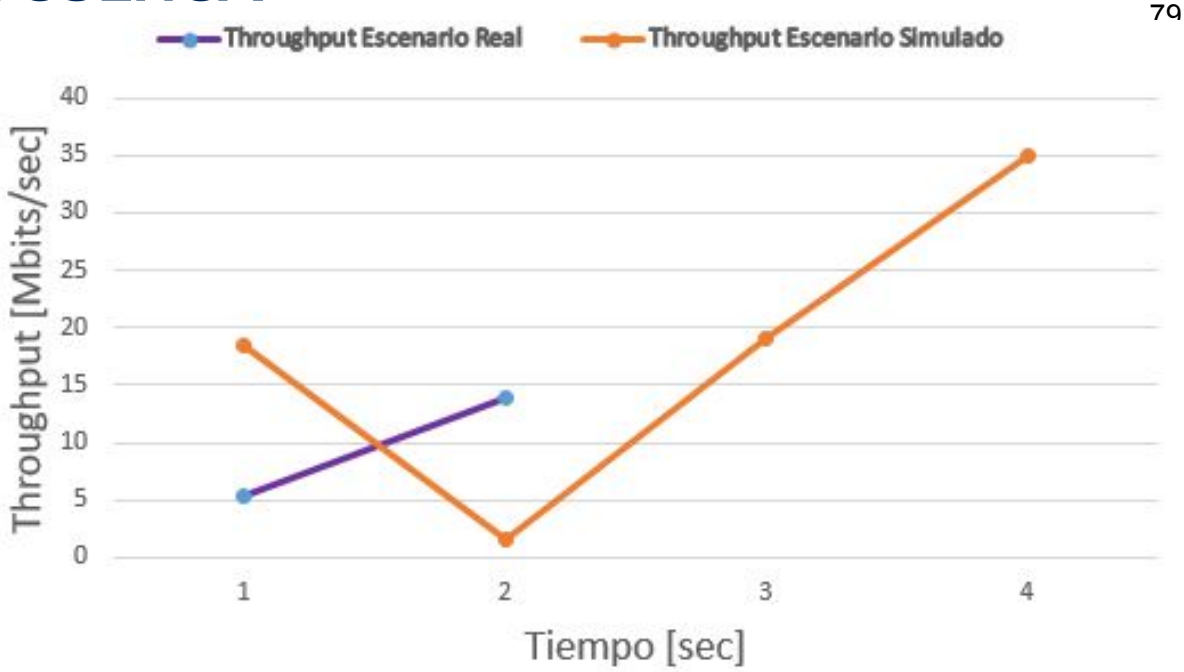



Figura 5.6: *Throughput* de nuclear transferido mediante TCP en escenario real y simulado

### 5.1.2. Ancho de Banda

El ancho de banda se obtuvo mediante el informe adquirido por Iperf3 desde el cliente en el cual se muestra al final el ancho de banda promedio durante el experimento (ver figura 5.7).
Para la mayoría de los experimentos se obtuvo un ancho de banda mayor en los escenarios reales, con excepción de cuando se transmitió una imagen nuclear en el cual el escenario simulado tuvo un ancho de banda superior el cual era de esperarse ya que el *throughput* también mejoró con respecto al escenario real.
La diferencia de los anchos de bandas obtenidos entre los escenarios reales y simulados para las imágenes de mastografía, ultrasonido y TAC no es tan significativa. En el caso de la resonancia muestra una diferencia de 15,74 Mbps en el escenario simulado. También se muestra como el ancho de banda en el escenario real se mantiene estable y en el caso del escenario simulado se tiene una degradación hasta el envío de la resonancia.
Para éste análisis se realizaron intervalos de confianza del 95 % para cada experimento. Como se muestra en la figura la mayoría de los experimentos muestran un intervalo de confianza pequeño excepto al transmitir las imágenes nuclear los cuales son más grandes lo que implica que los valores de ancho de banda son muy fluctuantes en esos experimentos.

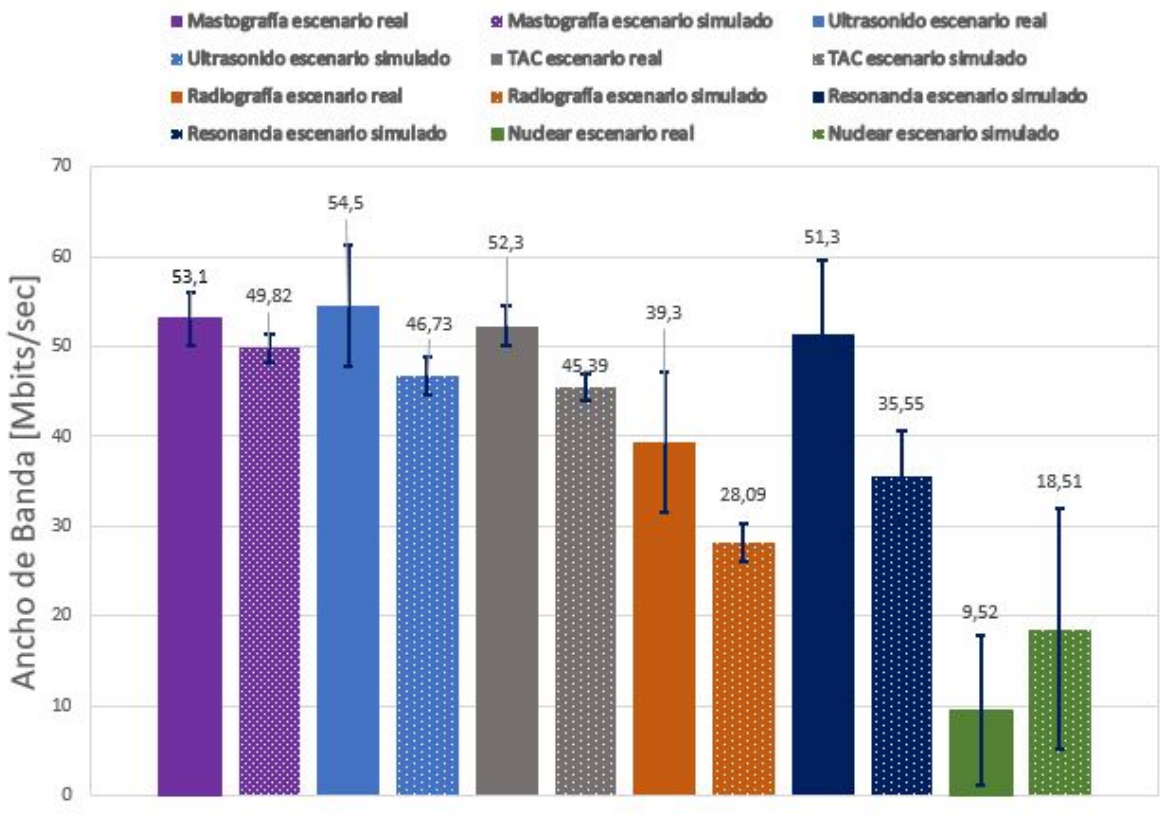
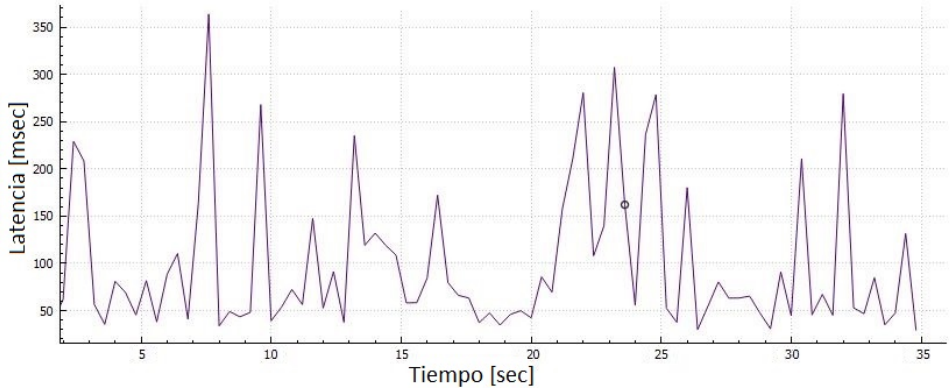


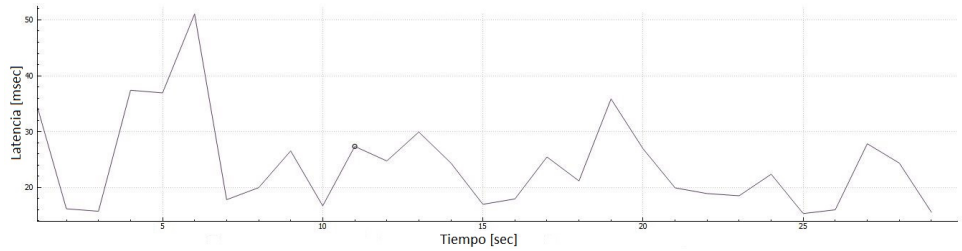

Figura 5.7: Anchos de Bandas promedio de todos las imágenes médicas transferidas mediante TCP en escenario real y simulado

### 5.1.3. Latencia

- **Mastografía**




(a) Latencia del escenario simulado




(b) Latencia del escenario real

Figura 5.8: Latencias del escenario real y simulado para una mastografía enviada mediante el protocolo TCP

Las gráficas que se muestran en la figuras 5.8a y 5.8b indican la latencia obtenida durante el experimento.

Para éste experimento la latencia se ve muy afectada en el escenario simulado ya que este muestra muchos picos lo que indica inestabilidad en el sistema, en cambio para el escenario real los valores medidos presentan mayor estabilidad.

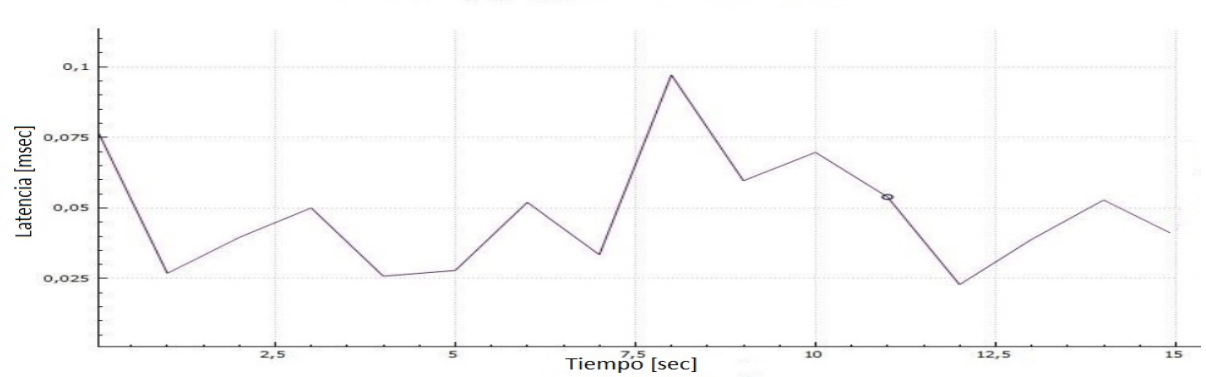
- **Ultrasonido de 60MB**

  Las gráficas que se muestran en la figura 5.9a y 5.9b indican el ancho de banda promedio obtenido durante el experimento..

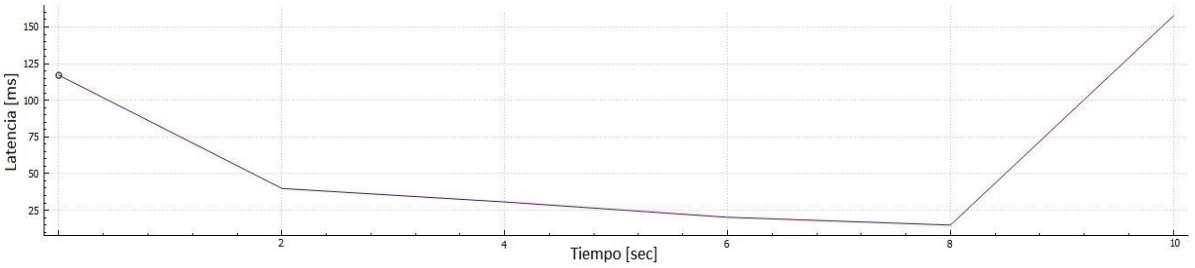
  En el escenario simulado se nota la mejora en la latencia a pesar de los picos que evidencia la gráfica ya que éste llega a 0,1 msec y en el real se tienen valores de hasta 125 msec.

- **TAC**

  Esta métrica se obtuvo mediante las herramientas de Wireshark para generar gráficas. Las gráficas que se muestran en las figuras 5.10a y 5.10b indican el ancho de banda promedio obtenido durante el experimento.




(a) Latencia del escenario simulado




(b) Latencia del escenario real

Figura 5.9: Latencias del escenario real y simulado para un ultrasonido de 60MB enviada mediante el protocolo TCP

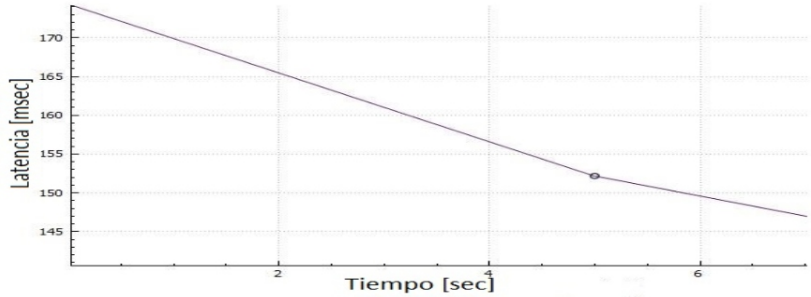
El tamaño del TAC implementado es de 20 MB lo que implica que es mucho menor, por lo que la latencia tiende a mejorar y ser más estable. En este experimento particular se puede ver como en el caso de la simulación la latencia tiende a descender por debajo de 150 msec, en cambio en el escenario real tiende a ascender sobre los 150 msec, por lo que no se puede obtener una conclusión clara de cual de los dos escenarios se comporta de mejor forma.

- **Radiografía**

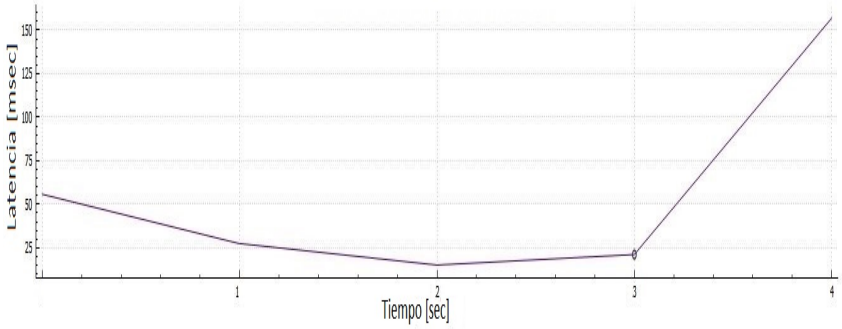
  La latencia obtenida para éste experimento muestra algunos picos e inestabilidad durante la transmisión en ambos escenarios, lo que se traduce en retraso en los paquetes; sin embargo, el valor pico de la latencia en el escenario simulado es de 600 msec, en cambio en el escenario real es de 800 msec lo que indica que en esos instantes de tiempo específicos hubo grandes retardos en la transmisión del archivo. Para éste experimento no se puede tener una conclusión clara; sin embargo, la transmisión del escenario real es más corto ya que no transmitió todos los paquetes.

- **Resonancia**

  Los valores obtenidos de latencia para el escenario simulado siguen siendo menores a




(a) Latencia del escenario simulado




(b) Latencia del escenario real

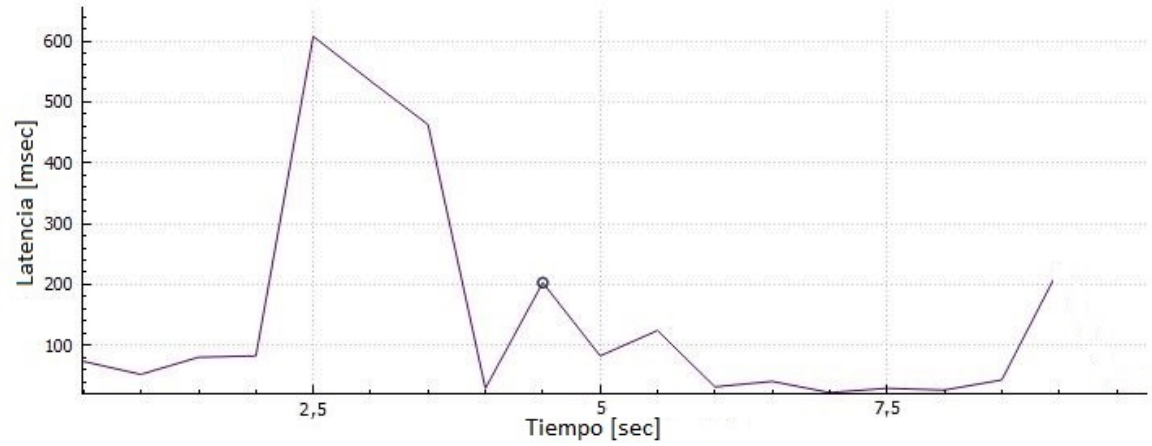
Figura 5.10: Latencias del escenario real y simulado para un TAC enviada mediante el protocolo TCP

los del real, lo que implica que se mantiene una mejora de ésta métrica.

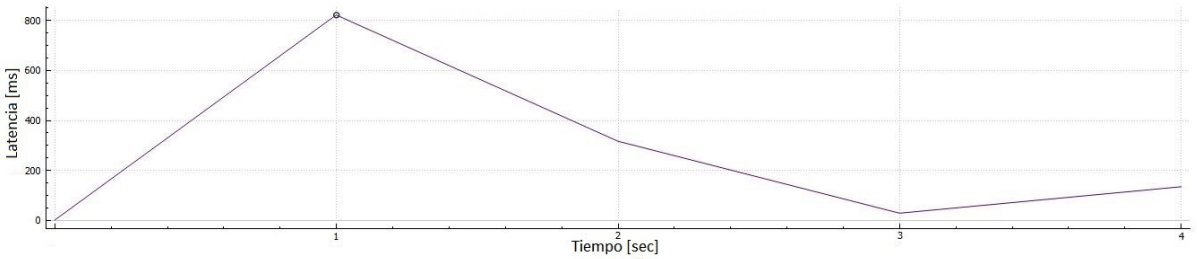
- **Nuclear**

  La métrica en este experimento mostró mejores resultados en el simulado ya que a pesar de que ambos tienden a disminuir el real tiene valores más alto.

Según los resultados obtenidos para la métrica de *throughput* al enviar paquetes mediante el protocolo de comunicación TCP se puede concluir que no se logró mejorar ésta métrica con las aplicaciones de SDN que se implementaron; sin embargo, se puede destacar que al usar archivos muy grandes muestra mejores rendimientos que a la hora de transmitir archivos de pequeños tamaños, desde el experimento dónde se transmitió un TAC que pesa 50 MB el *throughput* disminuyó progresivamente en los siguientes experimentos, pero también hubo una disminución de ésta métrica en el escenario real a medida que el archivo era más pequeño. Para la métrica de ancho de banda promedio en el caso de los experimentos realizados se obtuvieron valores altos al transmitir archivos grandes, para éstos escenarios se contaba con 100 MB en la interfaz física; sin embargo, no se logró un buen control de ésta métrica porque a medida que se disminuye el tamaño de los archivos el ancho de banda se ve afectado e incluso aún más que el escenario real, el cual también tuvo el mismo comportamiento.

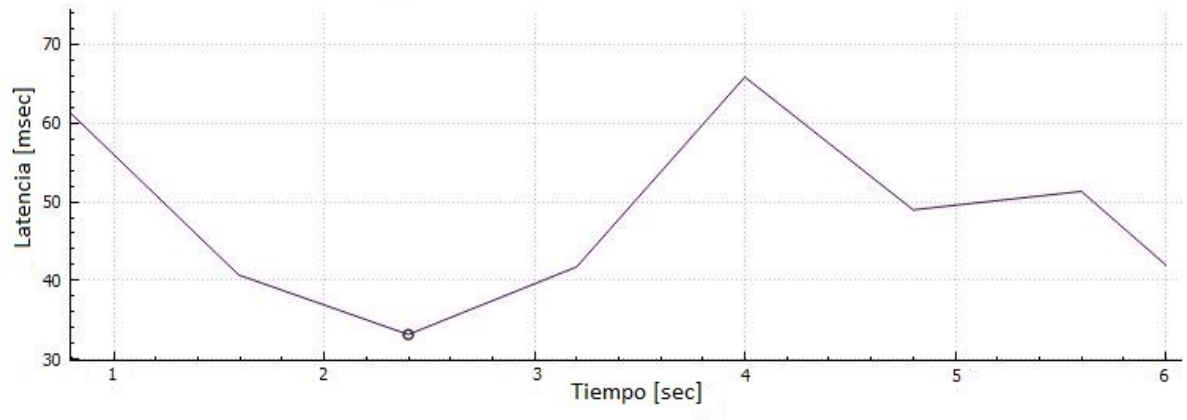



(a) Latencia del escenario simulado

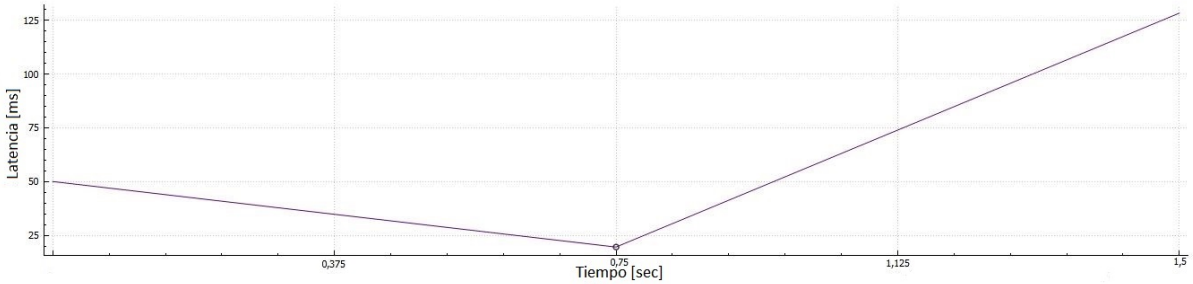



(b) Latencia del escenario real

Figura 5.11: Latencias del escenario real y simulado para una radiografía enviada mediante el protocolo TCP

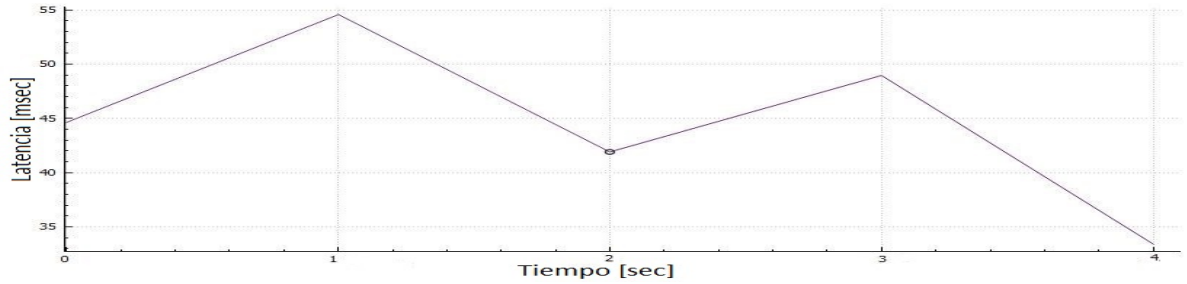



(a) Latencia del escenario simulado

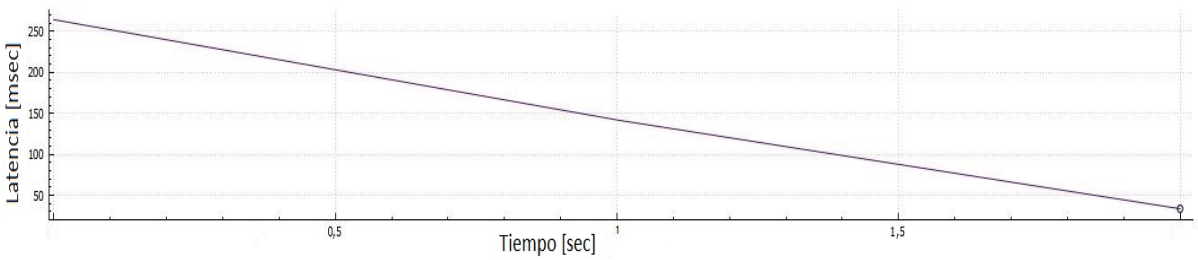



(b) Latencia del escenario real

Figura 5.12: Latencias del escenario real y simulado para una resonancia enviada mediante el protocolo TCP




(a) Latencia del escenario simulado




(b) Latencia del escenario real

Figura 5.13: Latencia del escenario real y simulado para una imagen nuclear enviada mediante el protocolo TCP

Los valores de la latencia obtenida para todos los experimentos usando el protocolo TCP, en el caso del escenario simulado fueron mejores que los obtenidos en el escenario real, a excepción del experimento transmitiendo una mastografía, el cual corresponde al archivo más grande, lo cual es un punto muy positivo dentro de éste análisis.

Es importante destacar una ventaja del escenario simulado sobre el escenario real, en éste se recibió la totalidad de los paquetes del archivo, a diferencia del escenario real, el cual para ninguna imagen se transmitió de manera completa.

## 5.2. Análisis de métricas de QoS de imágenes médicas utilizando el protocolo UDP

### 5.2.1. Ancho de Banda

Las gráficas que se muestran en las figuras 5.14 indican los diferentes anchos de bandas promedios obtenidos durante los experimentos. Se puede ver que los anchos de banda obtenidos en el escenario real para las diferentes imágenes médicas son mayor que los obtenidos en el escenario simulado a excepción del ultrasonido el cual se consiguió una mejora con respecto al escenario real. En la transmisión por UDP se puede ver como el escenario simulado controla mucho mejor el ancho de banda ya que no se tienen las degradaciones que se evidenciaron en el caso de TCP.

Para mastografía ambos escenario tienen muy buenos valores de anchos de banda ya que casi llegan a los 100 Mbps que se tienen disponible, sin embargo, el escenario real tiene mejor ancho de banda con 94,3 Mbps sobre los 92,53 Mbps obtenidos en la simulación, como se puede ver la diferencia entre ambas es despreciable. Por último, la diferencia que existe entre los anchos de banda simulados y reales es mínima.

Para éste análisis se realizaron intervalos de confianza del 95 % para cada experimento. Como se muestra en la figura 5.14 en todos los experimentos muestran un intervalo de confianza pequeño lo que implica que los valores de ancho de banda se encuentran alrededor de la media.

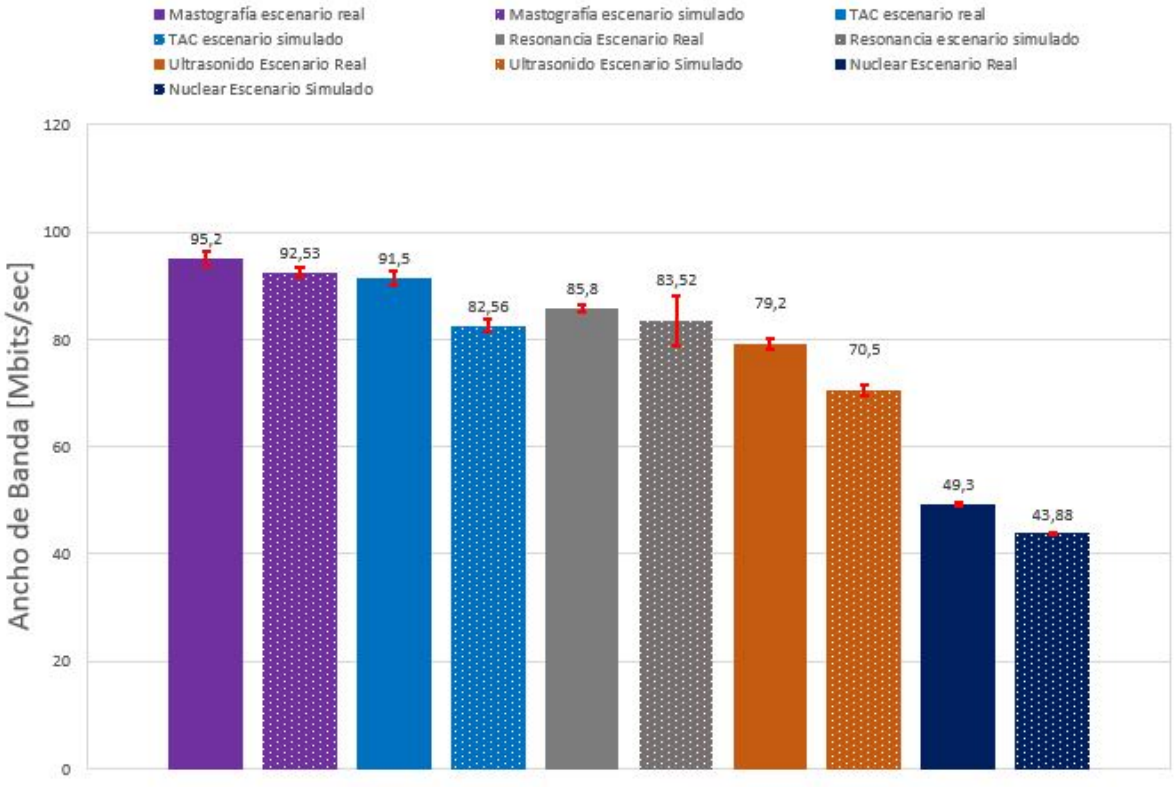



Figura 5.14: Anchos de Banda promedio de todas las imágenes médicas transferidas mediante UDP en escenario real y simulado

### 5.2.2. Pérdida de Paquetes

La pérdida de paquetes se obtuvo mediante el informe adquirido por Iperf3 desde el cliente en el cual se muestran la cantidad de paquetes transmitidos y la cantidad de paquetes que se recibieron durante todo el experimento. La gráfica 5.15 muestra el porcentaje de paquetes perdidos en ambos escenarios para todos las imágenes.

Para ésta métrica los diferentes experimentos en los escenarios simulados obtuvieron menor cantidad de paquetes perdidos lo que indica un control de ésta métrica, solo en la transferencia de la imagen nuclear, el escenario real fue menor al simulado ya que no tiene perdida de paquetes.

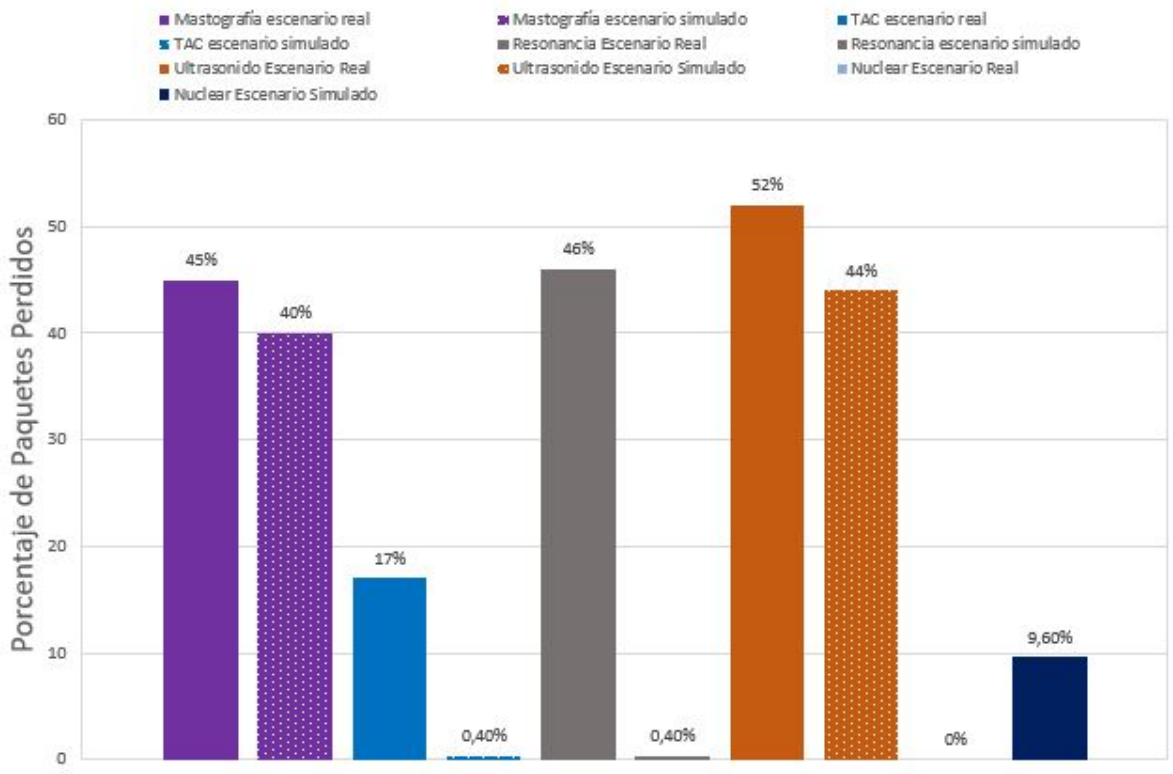



Figura 5.15: Porcentajes de paquetes perdidos de todas las imágenes médicas transferidas mediante UDP en escenario real y simulado

### 5.2.3. Jitter

El *jitter* se obtuvo mediante el informe adquirido por Iperf3 desde el cliente en el cual muestra un *jitter* promedio de todo el experimento. La figura 5.16 muestra todos los datos obtenidos tanto para los escenarios reales como simulados de todas las imágenes.

Todos los valores obtenidos en el *jitter* son aceptables para este tipo de escenario dónde se quieren enviar imágenes médicas, sin embargo, no se puede llegar a una conclusión certera ya que en algunos experimentos el *jitter* en el escenario simulado es más bajo que en el real; sin embargo, el *jitter* más alto entre todos los experimentos lo tuvo el escenario real cuando se transmitió una resonancia. También es notorio la mejoría de ésta métrica al transmitir archivos de mayor tamaño (50MB - 160MB).

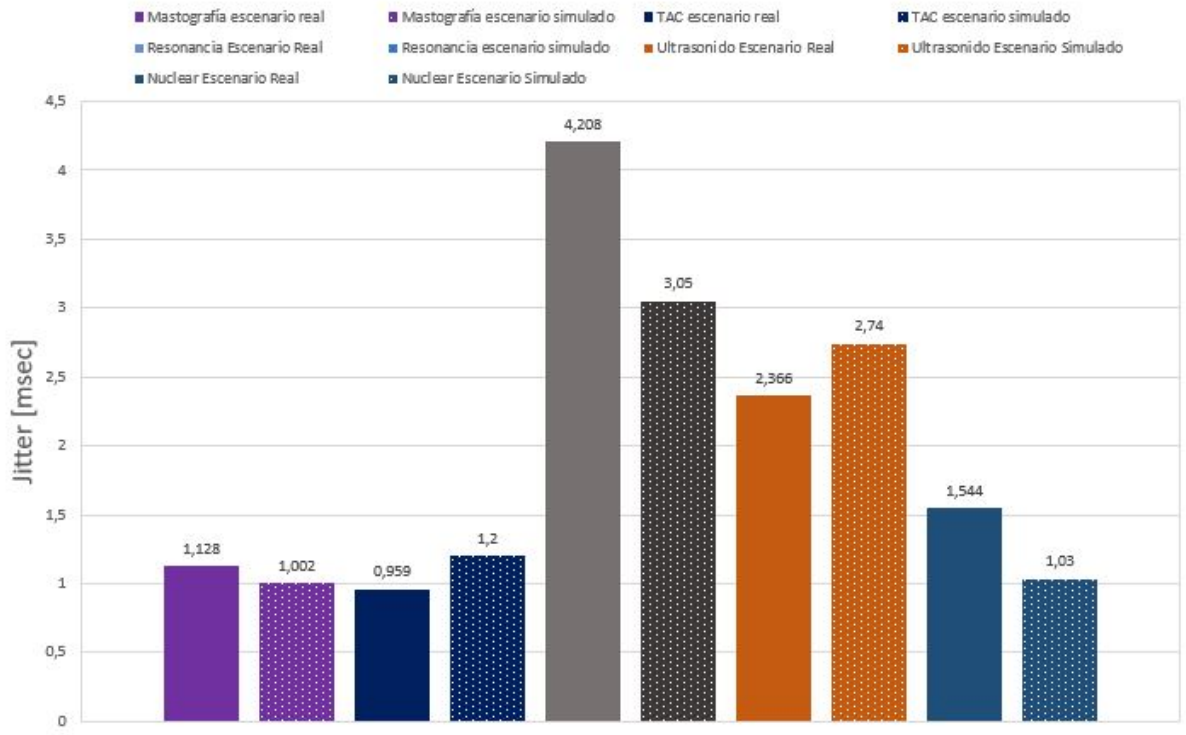



Figura 5.16: *Jitter* de todas las imágenes médicas transferidas mediante UDP en escenario real y simulado

### 5.2.4. Latencia

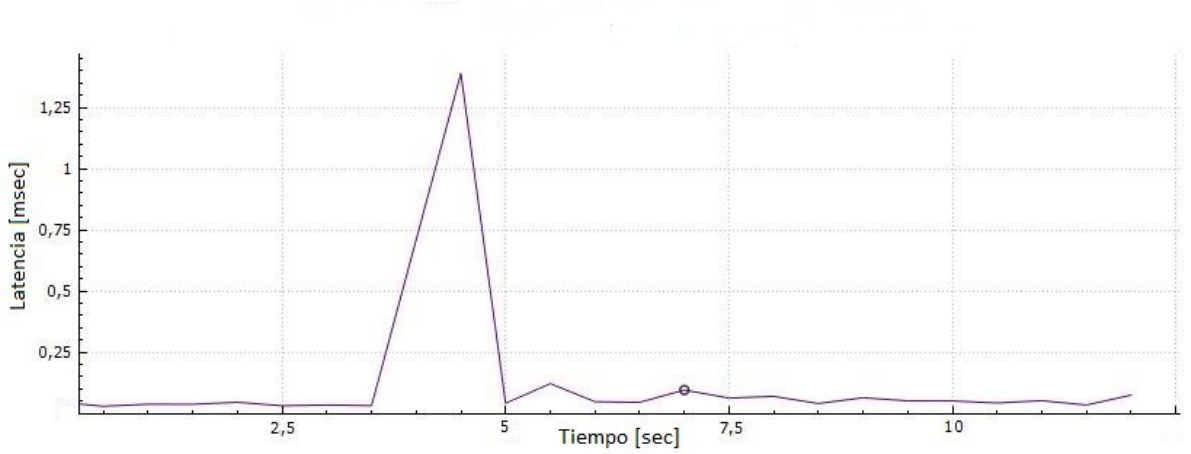
- **Mastografía.**

  Esta métrica se obtuvo mediante las herramientas de Wireshark para generar gráficas. Las gráficas que se muestran en las figuras 5.17a y 5.17b indican el ancho de banda promedio obtenido durante el experimento. Para este experimento se muestran como ambos escenarios tienen grandes picos en el experimento pero en el caso de la simulación solo se muestra un solo pico y se estabiliza caso contrario en el real muestra más picos durante el experimento.

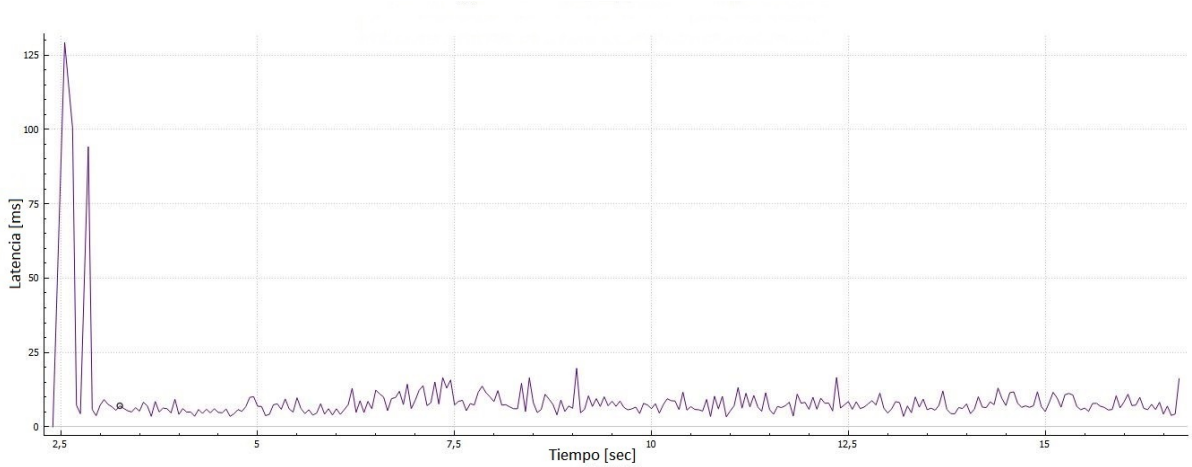
- **TAC.**

  Esta métrica se obtuvo mediante directamente de las herramientas de Wireshark para generar gráficas. Las gráficas que se muestran en las figuras 5.17a y 5.17b indican el ancho de banda promedio obtenido durante el experimento.

  En este experimento la métrica se comporta muy similar para ambos escenarios ya que a la ambos con una latencia elevada pero ambas tienden a descender a media que transcurre el experimento y luego vuelve a subir pero a valores aceptables. También se muestra que para el escenario simulado se tiene un pico más que en el escenario real, lo que se




(a) Latencia del escenario simulado

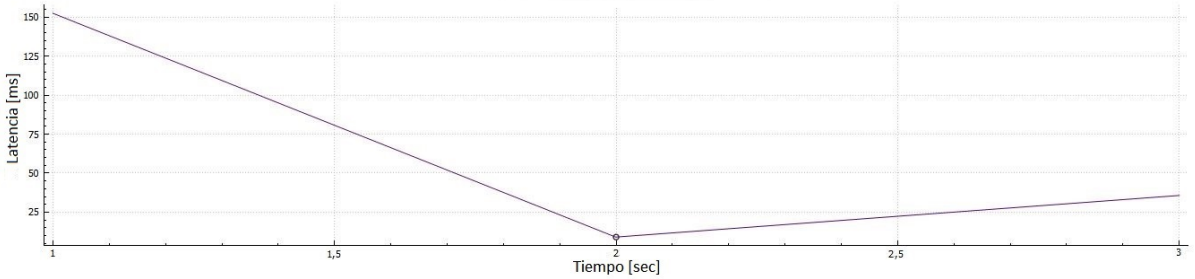



(b) Latencia del escenario real

Figura 5.17: Latencia del escenario real y simulado para un mastografía de 160MB enviada mediante el protocolo UDP

traduce en que ese instante de tiempo es probable que hubo un retraso en la transferencia de datos.


(a) Latencia del escenario simulado




(b) Latencia del escenario real

Figura 5.18: Latencia del escenario real y simulado para un TAC enviada mediante el protocolo UDP

- **Resonancia.**

  Esta métrica se obtuvo mediante de las herramientas de Wireshark para generar gráficas. Las gráficas que se muestran en las figuras 5.17a y 5.17b indican el ancho de banda promedio obtenido durante el experimento.

  Para este escenario se sigue manteniendo la mejoría de la latencia para el escenario simulado.

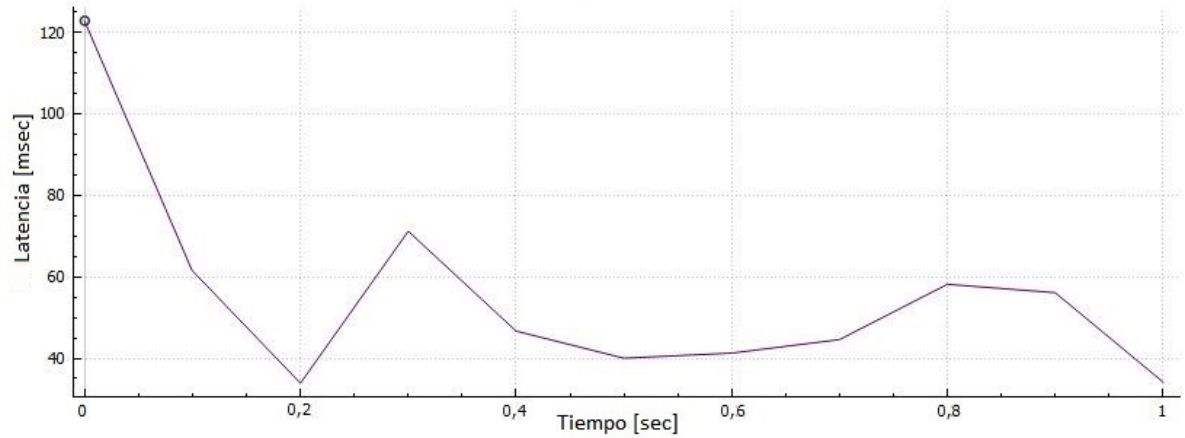
- **Ultrasonido de 5MB.**

  Para este experimento no se puede llegar a una conclusión certera ya que ambos tienen tendencia a disminuir su latencia pero ambos comienzan con una latencia muy similar.

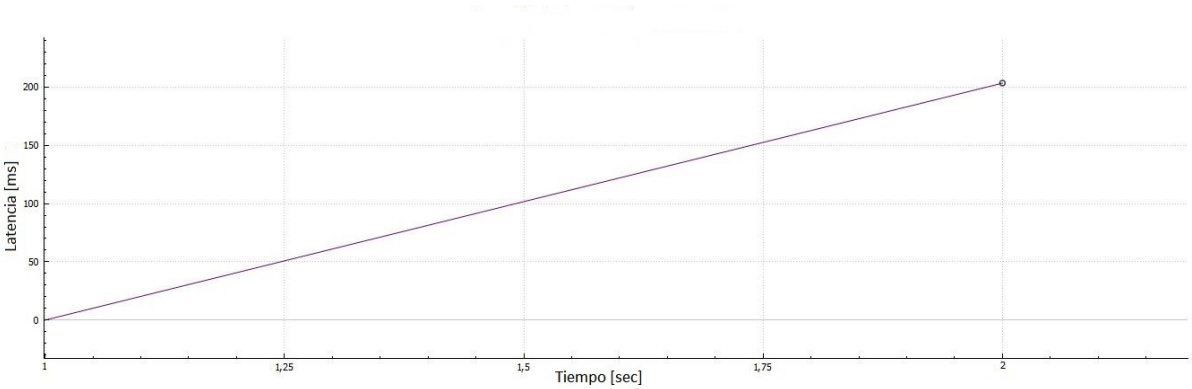
- **Nuclear de 1MB.**

  Es posible observar que el comportamiento de las latencias muestran una diferencia; sin embargo, las 2 tienden a reducir su valor, pero se consiguen mejores valores en el escenario simulado que en el escenario real.

Al principio con archivos grandes se obtuvieron valores de ancho de banda excelente para ambos escenarios, pero no se encontró relación entre el tamaño del paquete y el valor de ancho de banda promedio obtenido. No se logró un control de ésta métrica ya que no se estableció

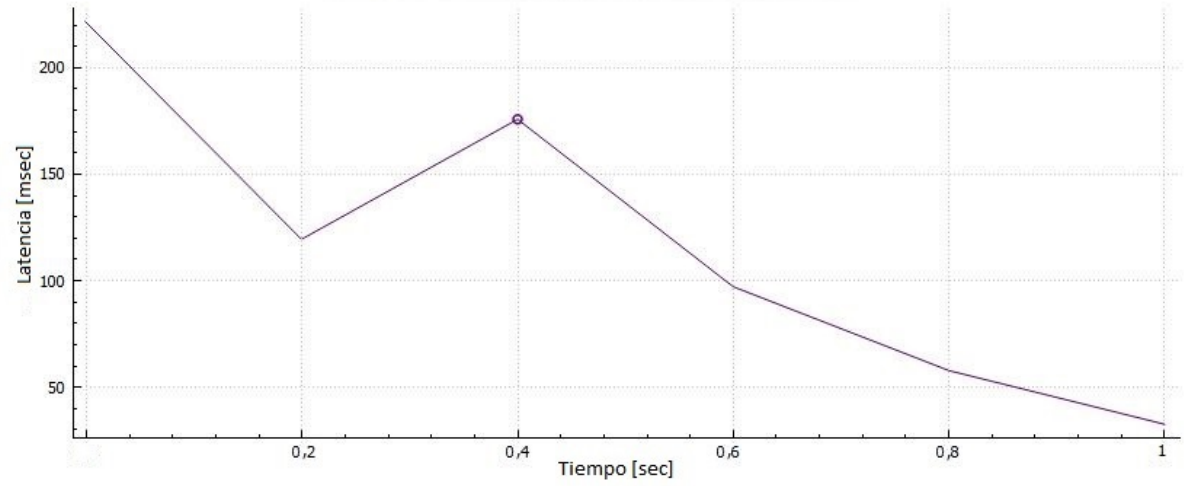



(a) Latencia del escenario simulado

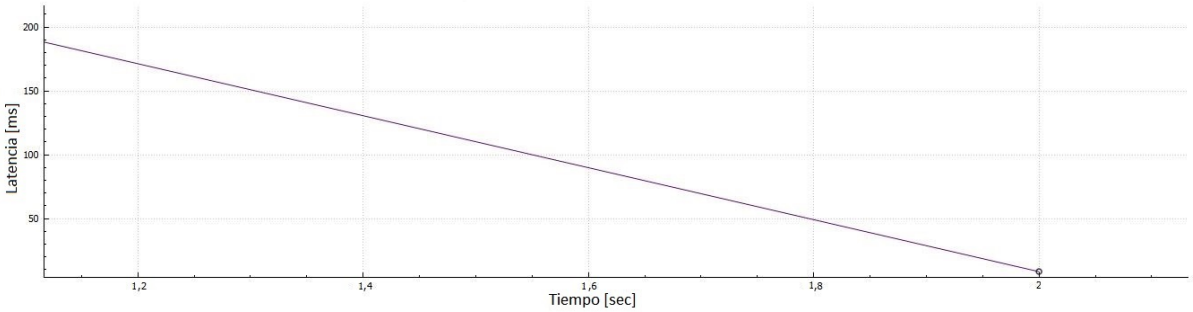



(b) Latencia del escenario real

Figura 5.19: Latencias del escenario real y simulado para una resonancia enviada mediante el protocolo UDP

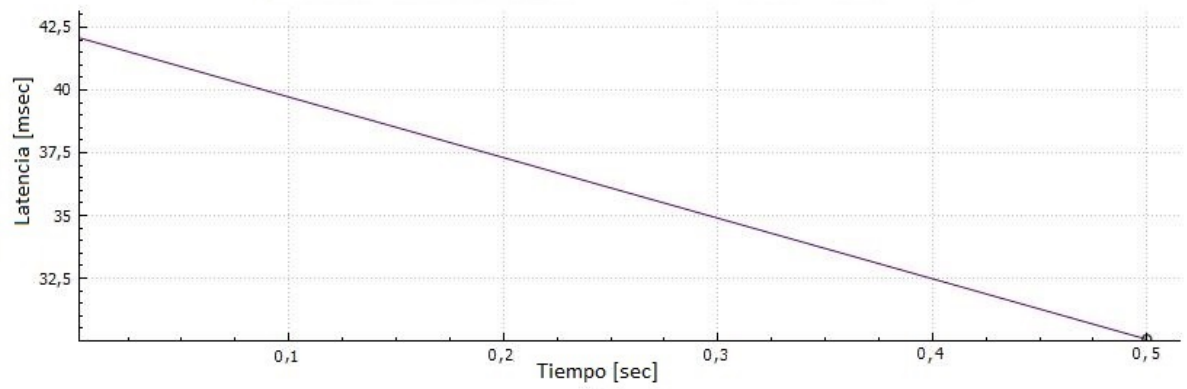



(a) Latencia del escenario simulado

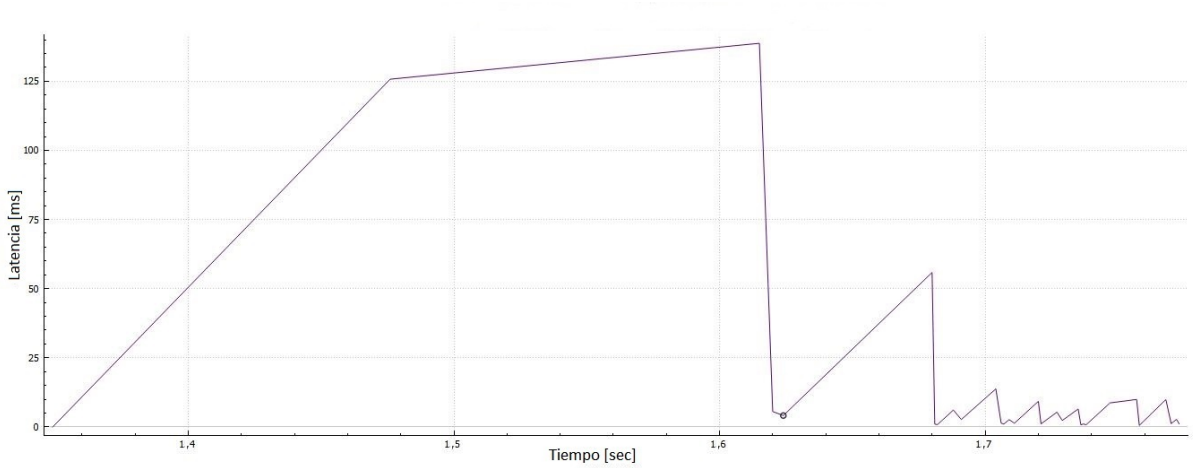



(b) Latencia del escenario real

Figura 5.20: Latencia del escenario real y simulado para un ultrasonido de 5MB enviada mediante el protocolo UDP




(a) Latencia del escenario simulado




(b) Latencia del escenario real

Figura 5.21: Latencia del escenario real y simulado para una imagen nuclear de 1MB enviada mediante el protocolo UDP

ningún tipo de configuración para ello. Sería conveniente configurar un control en las interfaces donde se encuentran conectados los equipos médicos para así asegurar un ancho de banda mínimo para éste tipo de tráfico, ésta configuración podría realizarse utilizando un mapeo de MAC o un mapeo de puertos, preferiblemente MAC ya que éstas son direcciones únicas y no importaría si se deba cambiar de lugar el equipo médico.

### 5.2.5. Problemas resueltos

Se presentaron problemas dentro de las diferentes etapas de este trabajo, las cuales a todas se intentaron encontrar alternativas con las que se pudiera cumplir con los objetivos propuestos. A la hora de realizar la caracterización del tráfico de red de la clínica, entre los experimentos generando tráfico con Iperf3 mediante el protocolo de comunicaciones TCP se observó que no se lograba transmitir todo el archivo, solo se lograban enviar entre un 30-40 %, sin importar el tamaño o la duración del experimento, por lo que se decidió generar nuevos archivos para lograr obtener en el cliente un archivo con las características de tamaño de las imágenes médicas ya seleccionadas. Entre las posibles causas ordenadas por orden de factibilidad se tienen:

- **Congestión:** en una red tradicional, la congestión podría ser causada por un alto volumen de tráfico en un enlace específico. Si la congestión se produce en el camino de la transferencia de archivos TCP, esto podría afectar negativamente la velocidad de transferencia y, en última instancia, la capacidad de completar la transferencia en el tiempo previsto. UDP, al no tener mecanismos de control de congestión, podría evitar este problema y lograr una transferencia completa (Bonaventure, 2011).
- **Firewalls y filtros de red:** tanto en el cliente como en el servidor, podrían bloquear el tráfico necesario para la transferencia de archivos, lo que resultaría en una transferencia incompleta. Algunos firewalls realizan inspección profunda de paquetes para analizar el contenido del tráfico. Si el firewall identifica un patrón que considera sospechoso o malicioso en los paquetes transmitidos, podría bloquear esos paquetes y esto podría ocurrir en protocolos como TCP. Otra posible causa son las políticas de seguridad del firewall pueden variar según la configuraciones. Por último, si el archivo que se intentó transferir está marcado por los filtros de contenido como potencialmente dañino o no permitido, el firewall podría bloquear la transferencia (Networks, 2023).
- **Política de Descartes de Paquetes en Router:** algunos routers o dispositivos de red pueden manejar y priorizar el tráfico TCP de manera diferente al tráfico UDP. Esto podría

hacer que los paquetes UDP sean tratados de manera más favorable en la red, lo que se traduce en una transferencia completa del archivo (Doyle y Carroll, 2001).

Es necesario tener en cuenta el espacio del disco disponible de las computadoras que se quieran utilizar, ya que es posible quedarse sin recursos luego de realizar todas las configuraciones necesarias, en especial si se desea trabajar con una topología grande. A pesar de contar con un servidor con características potentes en comparación a una computadora personal promedio, se hizo imposible el levantamiento de una topología más grande como la específica de la clínica, lo mismo ocurrió al querer encender todas las computadoras y lograr generar tráfico mediante Iperf3, existieron limitaciones computacionales. La figura 5.22 es la captura más próxima a que el sistema operativo dejara de responder; sin embargo, el consumo de recursos a la hora de realizar las simulaciones era aún mayor al mostrado en la figura, es posible ver que ambas máquinas virtuales consumen recursos de CPU, memoria y disco.

1. **Consumo de CPU:**
    - Un consumo de CPU del 52 % no se considera extremadamente alto, pero puede tener un impacto significativo en el rendimiento, especialmente ya que se encuentra a valores más altos durante un período prolongado.
    - Las aplicaciones que requieren potencia de procesamiento, como el simulador pueden experimentar retrasos o caídas en el rendimiento debido a la CPU ocupada.
    - Las aplicaciones multitarea también pueden verse afectadas, ya que la capacidad de la CPU para manejar múltiples procesos puede verse limitada, como es en el caso de Iperf3 el cual se ejecutan muchas instancias al mismo tiempo.
2. **Consumo de memoria:**
    - Un consumo de memoria del 62 % puede indicar que una cantidad considerable de memoria RAM está siendo utilizada por aplicaciones y procesos.
    - Las aplicaciones que utilizan mucha memoria, como el simulador, el controlador, navegadores, pueden experimentar retrasos en la carga y la respuesta.
3. **Consumo de disco:**
    - Un consumo de disco del 2 % es relativamente bajo y, por lo general, no debería ser la causa principal de ralentización. Sin embargo, si las operaciones de disco se vuelven intensivas, como la ejecución de Iperf3 en diferentes instancias, el porcentaje de uso del disco podría aumentar temporalmente y causar retrasos en otras operaciones.

En conjunto, una combinación de recursos de CPU, memoria y disco a los porcentajes que se muestran en la figura 5.22 podría llevar a una experiencia menos fluida y a un rendimiento más lento, especialmente al realizar tareas que demandan más recursos.

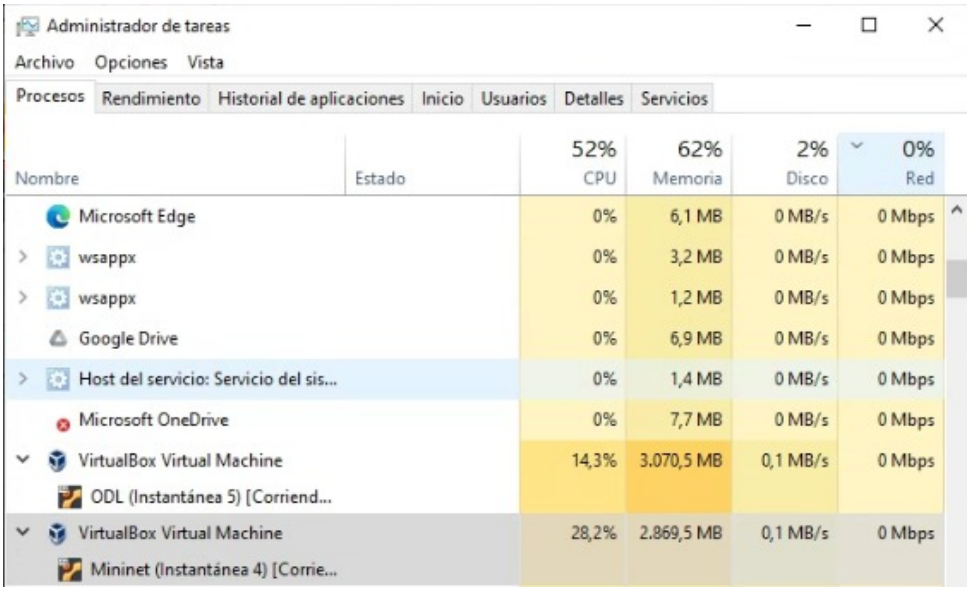



Figura 5.22: Uso de los recursos del servidor mediante la simulación

Se querían utilizar versiones más actualizadas tanto para MiniNet como para Opendaylight, pero la versión más actual para la fecha de MiniNet muestra errores cuando se quieren ejecutar scripts con Python3. En el caso del controlador, es necesario saber que componentes se desean utilizar para decidir de manera correcta que versión utilizar ya que no todas las versiones cuentan con los mismos componentes.

A pesar de contar con un servidor de 16 procesadores solo se pudo utilizar 8 procesadores para la máquina virtual con MiniNet ya que al aumentar la cantidad de procesadores el rendimiento de la máquina disminuyó mucho al punto que era imposible utilizarla porque tomaba mucho tiempo en abrir cualquier aplicación dentro de ésta.

# Conclusiones y Trabajos Futuros

## 6.1. Conclusiones

Con base en el desarrollo del trabajo de grado, centrado en la evaluación de métricas de QoS mediante la implementación de redes SDN para mejorar la transmisión de paquetes en telediagnóstico, hemos logrado alcanzar satisfactoriamente los objetivos planteados.

Se logró identificar una serie de métricas de QoS para escenarios de telediagnóstico, mediante un análisis exhaustivo y la revisión de la literatura especializada, se tuvo la intención de transmitir imágenes DICOM entre un escenario cliente-servidor, el cual emularía el escenario de telediagnóstico, pero al encontrar limitaciones en el simulador para transmitir imágenes, se decidió emular éstas imágenes utilizando archivos *randoms*, y por esa razón no se midieron también las métricas de PSNR y SSIM ya que no se tenían imágenes para calcular dichas métricas.

Se realizó la definición del escenario mediante un levantamiento de la topología de red con la que cuenta la clínica además de varias reuniones que se mantuvieron con la dirección del área de sistemas quienes brindaron información necesaria para llevar a cabo éste trabajo. La caracterización del tráfico de red de la clínica se logró mediante los experimentos con los diferentes archivos que emulan las imágenes médicas para luego obtener las métricas ya seleccionadas para éste trabajo. Esta etapa fue esencial para comprender las necesidades y particularidades del telediagnóstico y, así, adecuar el diseño de la red SDN para cumplir con los requisitos específicos de este escenario.

Se diseñó y configuró la topología de red para el escenario SDN basado en Python para luego ejecutarlo en el entorno de simulación MiniNet, con base en los datos y requerimientos recopilados. Luego, mediante el uso del controlador OpenDaylight se realizaron configuraciones específicas para lograr la mejora en la transmisión de paquetes para telediagnóstico.

Por último, se realizó un análisis comparativo de las métricas de QoS obtenidas en el escenario real y simulado. Se puede concluír que, utilizando el protocolo TCP se tenían muy buenos resultados tanto para el ancho de banda como para el *throughput* cuando se transmitieron imágenes de gran tamaño mientras se reducía el tamaño no se logró mantener un control de éstas métricas; sin embargo, se logró tener mejores valores de latencia en la mayoría de los experimentos en comparación a los obtenidos en el escenario real. Asimismo, existe un beneficio notorio al utilizar ésta tecnología para la transmisión de archivos grandes como las mastografías. Para un adecuado servicio de telediagnóstico éste tipo de archivos incremen-

tarán en la medida que dicho servicio sea más usado por la clínica. Por lo tanto, las redes tradicionales no podrán asegurar el buen rendimiento y funcionamiento de la transmisión de éstos archivos, mientras que con el uso de SDN se garantizará una transmisión con menor latencia. Y para el protocolo UDP, es necesario mencionar que al igual que con TCP se logró una mejora destacable en la latencia en la mayoría de los escenarios planteados. Considérese que en telemedicina general y en telediagnóstico en especial la latencia puede ser un factor que afecte de manera crítica a las aplicaciones tales como, telecirugía. Además, es claro que el ancho de banda en éste protocolo puede mantenerse estable y cercano al límite de la interfaz física, así como la métrica de la pérdida de paquetes se tuvo un mejor resultado con el uso de la tecnología SDN por lo tanto una mejora en la QoS para éste tipo de aplicación.

Finalmente, este trabajo de titulación ha contribuido al avance del conocimiento en el área de telediagnóstico y redes SDN, demostrando su viabilidad y beneficios en la mejora de la QoS en esta aplicación específica. Los resultados obtenidos pueden ser de gran utilidad para profesionales de redes, ingenieros de telecomunicaciones y aquellos interesados en la implementación de tecnologías innovadoras para mejorar la transmisión de datos en aplicaciones de telemedicina. Así mismo, la configuración de los simuladores con la complejidad de la topología tiene una sección de problemas resueltos que puede aportar a quienes deseen utilizar los simuladores que se describieron en éste trabajo.

### 6.2. Trabajos Futuros

Se sugiere utilizar módulos dentro del controlador para configurar el ancho de banda en las interfaces dónde se encuentren conectados los equipos médicos, o configurar un mapeo de MAC para que ésta característica solo sea aplicada al equipo médico sin importar su ubicación física dentro de la torre, también se podrían realizar configuraciones más específicas dónde el controlador recopile los valores de ciertas métricas por un tiempo establecido, y fijar valores críticos y valores estables, si alguna métrica llega a los valores críticos enviar una notificación al ingeniero en telecomunicaciones y active algunas configuraciones extras que sirvan para que los valores de las métricas vuelvan a los valores estables.

# Anexos

## A.1. Tablas para la determinación de las métricas de QoS

| Número | Título | DOI |
|---|---|---|
| 1 | EFSUTE: a novel efficient and survivable traffic engineering for software defined networks | 10.1007/s40860-021-00139-0 |
| 2 | A Reinforcement Learning-Based Routing for Real-Time Multimedia Traffic Transmission over Software-Defined Networking | 10.3390/electronics11152441 |
| 3 | QoS Support Path Selection for Inter-Domain Flows Using Effective Delay and Directed Acyclic Graph in Multi-Domain SDN | 10.3390/electronics11142245 |
| 4 | Integrating Deep Learning-Based IoT and Fog Computing with Software-Defined Networking for Detecting Weapons in Video Surveillance Systems | 10.3390/s22145075 |
| 5 | Minimize the delays in software defined network switch controller communication | 10.1002/cpe.5940 |
| 6 | A Deep Learning Based Method for Network Application Classification in Software Defined IoT | 10.1142/S0218488522400165 |
| 7 | Video Streaming Adaptive QoS Routing with Resource Reservation (VQoSRR) Model for SDN Networks | 10.3390/electronics11081252 |
| 8 | Video transfer utilizing software defined network (SDN): A survey | 10.1063/5.0066845 |
| 9 | Dynamic Multipath Routing Mechanism for Multimedia Data Flow Scheduling Over Software Defined Networks | 10.1109/DSN-W54100.2022.00040 |

| Número | Título | DOI |
|---|---|---|
| 10 | Reinforcement Learning Based Routing in Software Defined Network | 10.1007/978-981-19-1018-0_16 |
| 11 | LARIC: latency-aware QoS routing for interactive communication in software defined multimedia | 10.1007/s41870-022-01053-1 |
| 12 | Resilience of Delay-sensitive Services with Transport-layer Monitoring in SD-WAN | 10.1109/TNSM.2022.3191943 |
| 13 | A CLASS-BASED ADAPTIVE QOS CONTROL SCHEME ADOPTING OPTIMIZATION TECHNIQUE OVER WLAN SDN ARCHITECTURE | 10.5121/ijcnc.2022.14304 |
| 14 | SDN Control Strategy and QoS Optimization Simulation Performance Based on Improved Algorithm | 10.1155/2022/7167957 |
| 15 | QoS-Aware Multicast for Crowdsourced 360° Live Streaming in SDN Aided NG-EPON | 10.1109/ACCESS.2022.3144477 |
| 16 | Implementation of an Efficient Handover Algorithm on AP Using Software-Defined Wi-Fi Systems | 10.1007/978-981-16-2761-3_105 |
| 17 | Intelligent Traffic Engineering for Future Intent-Based Software-Defined Transport Network | 10.1007/978-3-030-92435-5_9 |
| 18 | A QoS-Based routing algorithm over software defined networks | 10.1016/j.jnca.2021.103215 |
| 19 | A Software-Defined Queuing Framework for QoS Provisioning in 5G and beyond Mobile Systems | 10.1109/MNET.011.2000441 |
| 20 | A quality of experience management method for intent-based software-defined networks | 10.1109/CADSM52681.2021.9385250 |

| Número | Título | DOI |
|---|---|---|
| 21 | ML-based Application Performance Modelling over a SDN Network Using End-to-end and Link Metrics | 978-380075673-5 |
| 22 | Video Streaming Service Identification on Software-Defined Networking | 10.15837/ijccc.2021.5.4258 |
| 23 | A QoS-aware Routing based on Bandwidth Management in Software-Defined IoT Network | 10.1109/MASS52906.2021.00082 |
| 24 | Analysis of ONOS Clustering Performance on Software Defined Network | 10.1109/IoTaIS53735.2021.9628659 |
| 25 | Scalable QoS-Aware Multicast for SVC Streams in Software-Defined Networks | 10.1109/ISCC53001.2021.9631505 |
| 26 | A deep learning based traffic classification in software defined networking Networks | 978-989870427-6 |
| 27 | Effective Handover Process for Reliable Video Streaming Over Software-Defined Wireless Networking | 10.1007/978-3-030-72792-5_37 |
| 28 | Exploring the Effects of Network Topology Layers on Quality of Service Mechanisms in the Context of Software-Defined Networking | 10.1007/978-981-33-6385-4_2 |
| 29 | SMOTE: An Intelligent SDN-Based Multi-Objective Traffic Engineering Technique for Telesurgery | 10.1080/03772063.2021.1894248 |
| 30 | A Novel Traffic Shaping Algorithm for SDN-Sliced Networks using a New WFQ Technique | 10.14569/IJACSA.2021.0120101 |
| 31 | Fast bandwidth-delay routing methods for software-defined network (SDN) | 10.1007/978-981-15-5329-5_31 |
| 32 | Congestion control proposal in SDN with random early detection | 10.1109/ARGENCON49523.2020.9505361 |
| 33 | Fault tolerance in SDN data plane considering network and application based metrics | 10.1016/j.jnca.2020.102780 |

| Número | Título | DOI |
|---|---|---|
| 34 | Provision of tactile internet traffic in IP access networks using SDN-enabled multipath routing | 10.1109/ICT49546.2020.09239437 |
| 35 | Comparison of routing protocol performance on multimedia services on software defined network | 10.11591/eei.v9i4.2389 |
| 36 | A consistent QoS routing strategy for video streaming services in SDN networks | 10.1002/dac.4177 |
| 37 | Timber: An SDN-Based Emulation Platform for Experimental Research on Video Streaming | 10.1109/JSAC.2020.2999683 |
| 38 | Performance evaluation of routing strategies over multimedia-based SDNs under realistic environments | 10.1109/NetSoft48620.2020.9165521 |
| 39 | A QoS Model for Real-Time Application in Wireless Network Using Software Defined Network | 10.1007/s11277-020-07089-5 |
| 40 | Development of OpenFlow Native Capabilities to optimize QoS | 10.1109/SDS49854.2020.9143890 |
| 41 | Software defined network of video surveillance system based on enhanced routing algorithms | 10.21123/BSJ.2020.17.1(SUPPL.).0391 |
| 42 | A robust multimedia traffic SDN-Based management system using patterns and models of QoE estimation with BRNN | 10.1016/j.jnca.2019.102498 |
| 43 | Architecture and protocol to optimize videoconference in wireless networks | 10.1155/2020/490342 |
| 44 | QoS-Based routing algorithm for software-defined network using ant colony optimization | 10.1007/978-981-15-3338-9_5 |
| 45 | Design and validation of a meter band rate in OpenFlow and OpenDaylight for optimizing QoS | 10.25046/aj050205 |

| Número | Título | DOI |
|---|---|---|
| 46 | A QoS-Based Flow Assignment for Traffic Engineering in Software-Defined Networks | 10.1007/978-3-030-15032-7_64 |
| 47 | QoS-aware traffic engineering in software defined networks | 10.1109/APCC47188.2019.9026524 |
| 48 | AROMA: An Adapt-or-Reroute Strategy for Multimedia Applications Engineering in Software-Defined Networks | 10.1109/BMSB47279.2019.8971842 |
| 49 | Data collection in SDN networks with contextual analysis | 10.23919/CISTI.2019.8760999 |
| 50 | An SDN-based framework for improving the performance of underprovisioned IP Video Surveillance networks | 10.1016/j.jnca.2019.01.026 |
| 51 | A meter band rate mechanism to improve the native QoS capability of openflow and opendaylight | 10.1109/COMMNET.2019.8742360 |
| 52 | QoS improvisation of delay sensitive communication using SDN based multipath routing for medical applications | 10.1016/j.future.2018.10.032 |
| 53 | QoS-aware Resource Reallocation Based on Flow Priority in Software Defined Network | 10.1109/ICCChinaW.2018.8674501 |
| 54 | MaxStream: SDN-based Flow Maximization for Video Streaming with QoS Enhancement | 10.1109/LCN.2018.8638033 |
| 55 | An SDN-based framework for improving the performance of Underprovisioned IP video surveillance networks | 10.1109/I-SPAN.2018.00033 |
| 56 | Latency over software defined network | 10.1007/978-3-030-11928-7_48 |
| 57 | A Survey on Autonomic Provisioning and Management of QoS in SDN Networks | 10.1109/ACCESS.2019.2919957 |
| 58 | SSIM and ML based QoE enhancement approach in SDN context | 10.1016/bs.adcom.2019.02.004 |

| Número | Título | DOI |
|---|---|---|
| 59 | DTE-SDN: A Dynamic Traffic Engineering Engine for Delay-Sensitive Transfer | 10.1109/JIOT.2018.287243 |
| 60 | A proactive flow admission and re-routing scheme for load balancing and mitigation of congestion propagation in SDN data plane | 10.5121/ijcnc.2018.10607 |
| 61 | Scalable QoE-aware Path Selection in SDN-based Mobile Networks | 10.1109/INFOCOM.2018.8486427 |
| 62 | Fuzzy Logic Approach to Increase Quality A of Service in Software Defined Networking | 10.1109/ICACCCN.2018.8748678 |
| 63 | Admission control implementation for qos performance evaluation over SDWN | 10.1109/ColComCon.2018.8466339 |
| 64 | OQR: On-demand QoS Routing without Traffic Engineering in Software Defined Networks | 10.1109/NETSOFT.2018.8460088 |
| 65 | On the feasibility of telesurgery over software defined networks | 10.1007/s41315-018-0059-5 |
| 66 | Network optimization for differentiated QoS traffic in an SDN environment for PoP-data center traffic | 10.1109/SARNOF.2018.8720505 |
| 67 | S5: An Application Sensitive QoS Assurance System via SDN | 10.1109/PCCC.2018.8710838 |
| 68 | S5: IARA: An intelligent application-aware VNF for network resource allocation with deep learning | 10.1109/SAHCN.2018.8397153 |
| 69 | Network resource allocation system for QoE-aware delivery of media services in 5G networks | 10.1109/TBC.2018.2828608 |
| 70 | Dynamic QoS support for IoT backhaul networks through SDN | 10.1109/FMEC.2018.8364063 |
| 71 | The hardware and software co-design of a configurable QoS for video streaming based on OpenFlow protocol and NetFPGA platform SDN | 10.1007/s11042-017-4806-7 |

| Número | Título | DOI |
|---|---|---|
| 72 | An SDN framework for video conference in inter-domain network | 10.23919/ICACT.2018.8323848 |
| 73 | QoS-aware traffic classification architecture using machine learning and deep packet inspection in SDNs | 10.1016/j.procs.2018.04.331 |
| 74 | An intelligent traffic engineering method for video surveillance systems over software defined networks using ant colony optimisation | 10.1504/IJBIC.2018.094625 |
| 75 | Data Center Traffic Engineering: Multipath Routing with QoS Guarantee | 10.1007/978-981-10-7901-6_93 |

Cuadro A.1: Primer resultado del análisis bibliográfico

| Número | Título | DOI | Métricas de QoS |
|---|---|---|---|
| 1 | EFSUTE: a novel efficient and survivable traffic engineering for software defined networks | 10.1007/s40860-021-00139-0 | Avarage end-to-end delay<br>Jitter<br>Packet loss ratio<br>Path calculation time<br>Recovery time |
| 2 | A Reinforcement Learning-Based Routing for Real-Time Multimedia Traffic Transmission over Software-Defined Networking | 10.3390/electronics 11152441 | Link delay<br>Jitter<br>Loss ratio<br>Ancho de banda disponible |
| 3 | Integrating Deep Learning-Based IoT and Fog Computing with Software-Defined Networking for Detecting Weapons in Video Surveillance Systems | 10.3390/s22145075 | Delay<br>Throughput<br>Bandwidth |
| 4 | Resilience of Delay-sensitive Services with Transport-layer Monitoring in SD-WAN | 10.1109/TNSM.2022.3191943 | Packet Delay<br>Loss<br>Jitter<br>Service Requirments |
| 5 | A CLASS-BASED ADAPTIVE QOS CONTROL SCHEME ADOPTING OPTIMIZATION TECHNIQUE OVER WLAN SDN ARCHITECTURE | 10.5121/ijcnc.2022.14304 | Avarage Throughput<br>Latency<br>Jitter<br>Packet Loss Rate |
| 6 | SDN Control Strategy and QoS Optimization Simulation Performance Based on Improved Algorithm | 10.1155/2022/7167957 | No indica |

| Número | Título | DOI | Métricas de QoS |
|---|---|---|---|
| 7 | QoS-Aware Multicast for Crowdsourced 360°Live Streaming in SDN Aided NG-EPON | 10.1109/ACCESS.2022.3144477 | Mean packet delay<br>Jitter<br>System Throughput<br>Packet loss ratio<br>Queue lenght |
| 8 | A QoS-Based routing algorithm over software defined networks | 10.1016/j.jnca.2021.103215 | Network Latency |
| 9 | A quality of experience management method for intent-based software defined networks | 10.1109/CADSM52681.2021.9385250 | Delay<br>Packet loss |
| 10 | Video Streaming Service Identification on Software Defined Networking | 10.15837/ijccc.2021.5.4258 | No indica |
| 11 | A QoS-aware Routing based on Bandwidth Management in Software-Defined IoT Network | 10.1109/MASS52906.2021.00082 | End to end delay<br>Average Throughput<br>Average Jitter |
| 12 | A deep learning based traffic classification in software defined networking Networks | 978-989870427-6 | No indica |
| 13 | SMOTE: An Intelligent SDN Based Multi-Objective Traffic Engineering Technique for Telesurgery | 10.1080/03772063.2021.1894248 | Average end to end delay<br>Packet loss ratio<br>Jitter<br>PSNR<br>SSIM |

| Número | Título | DOI | Métricas de QoS |
|---|---|---|---|
| 14 | A Novel Traffic Shaping Algorithm for SDN-Sliced Networks using a New WFQ Technique | 10.14569/IJACSA.2021.0120101 | Throughput<br>Delay<br>Jitter |
| 15 | Fault tolerance in SDN data plane considering network and application based metrics | 10.1016/j.jnca.2020.102780 | Throughput<br>Packet Loss<br>Jitter<br>Recovery Loss. |
| 16 | Provision of tactile internet traffic in IP access networks using SDN-enabled multipath routing | 10.1109/ICT49546.2020.09239437 | End to end delay<br>Throughput del sistema |
| 17 | Comparison of routing protocol performance on multimedia services on software defined network | 10.11591/eei.v9i4.2389 | Throughtput<br>Delay<br>Jitter<br>Convergence Time<br>MOS |
| 18 | Timber: An SDN-Based Emulation Platform for Experimental Research on Video Streaming | 10.1109/JSAC.2020.2999683 | Throughput<br>Packet Loss<br>Delay |
| 19 | A QoS Model for Real-Time Application in Wireless Network Using Software Defined Network | 10.1007/s11277-020-07089-5 | Throughput<br>Packet Loss |
| 20 | Software defined network of video surveillance system based on enhanced routing algorithms | 10.21123/BSJ.2020.17.1(SUPPL.).0391 | End to end delay<br>Packet Loss Ratio<br>Peak Signal to Noise Ratio |

| Número | Título | DOI | Métricas de QoS |
|---|---|---|---|
| 21 | A robust multimedia traffic SDN-Based management system using patterns and models of QoE estimation with BRNN | 10.1016/j.jnca.2019.102498 | Delay<br>Jitter<br>Loss rate<br>Bandwidth |
| 22 | Architecture and protocol to optimize videoconference in wireless networks | 10.1155/2020/490342 | Packet Loss<br>End to end delay<br>Jitter |
| 23 | SSIM and ML based QoE enhancement approach in SDN context | 10.1016/bs.adcom.2019.02.004 | Jitter<br>RTT<br>Bandwidth. |
| 24 | On the feasibility of telesurgery over software defined networks | 10.1007/s41315-018-0059-5 | Average end to end delay<br>Packet loss ratio<br>Jitter<br>PSNR<br>SSIM<br>Average response time |
| 25 | QoS-Based routing algorithm for software-defined network using ant colony optimization | 10.1007/978-981-15-3338-9-5 | End to end delay<br>Packet Delivery Ratio<br>Control Overhead<br>Throughput |
| 26 | QoS-aware traffic engineering in software defined networks | 10.1109/APCC47188.2019.9026524 | Packet Loss Rate<br>Throughput |
| 27 | QoS improvisation of delay sensitive communication using SDN based multipath routing for medical applications | 10.1016/j.future.2018.10.032 | Delay time<br>Throughput |
| 28 | A Survey on Autonomic Provisioning and Management of QoS in SDN Networks | 10.1109/ACCESS.2019.2919957 | No indica |

| Número | Título | DOI | Métricas de QoS |
|---|---|---|---|
| 29 | An agent-based QoE monitoring strategy for LTE networks | 10.1109/ICC.2018.8422608 | No indica |
| 30 | LearnQoS: A Learning Approach for Optimizing QoS over Multimedia-based SDNs | 10.1109/BMSB.2018.8436781 | PSNR<br>SSIM<br>MOS<br>Throughput<br>Packet Loss Rate<br>Latency |

Cuadro A.2: Métricas de QoS utilizadas en los artículos del análisis bibliográfico

## A.2. Imágenes varias

Figura A.1: Cableado desde el Data Center hasta los racks de distribución de cada piso

## A.3. Informes de Iperf3

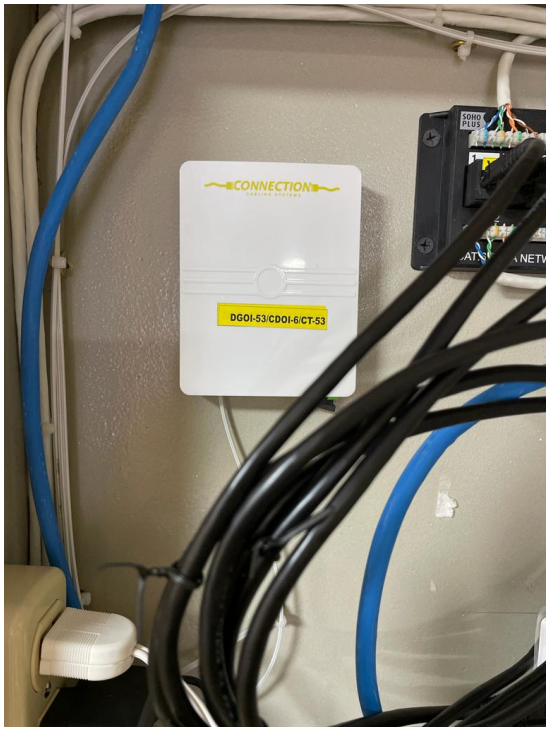



Figura A.2: Caja Final


Figura A.3: Rack de Distribución


Figura A.4: Rack del Data Center

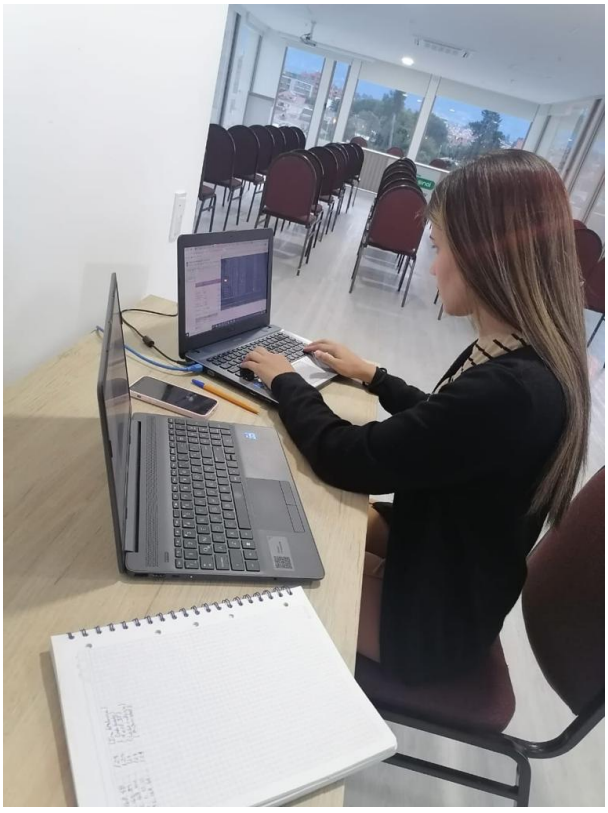


Figura A.5: Ejecución de pruebas en el escenario real

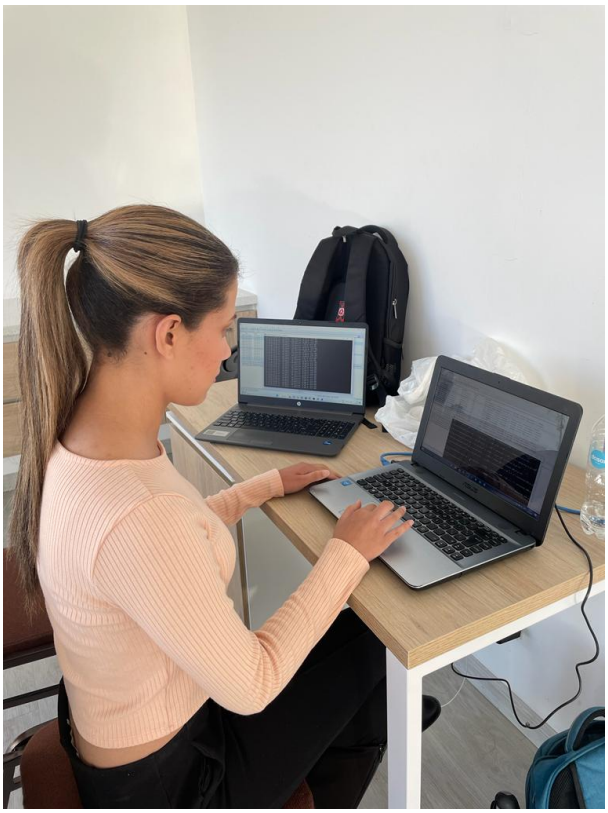


Figura A.6: Ejecución de pruebas en el escenario real

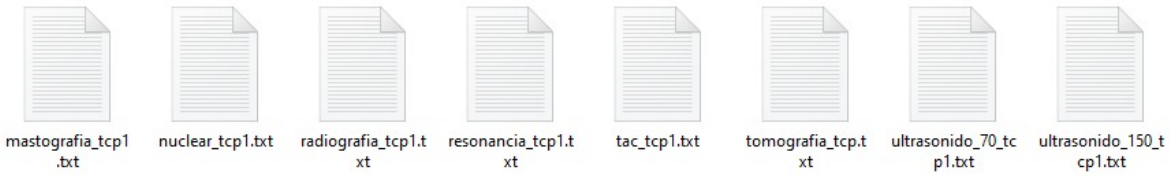



Figura A.7: Informes de resultados para todas las imágenes médicas utilizando el protocolo TCP

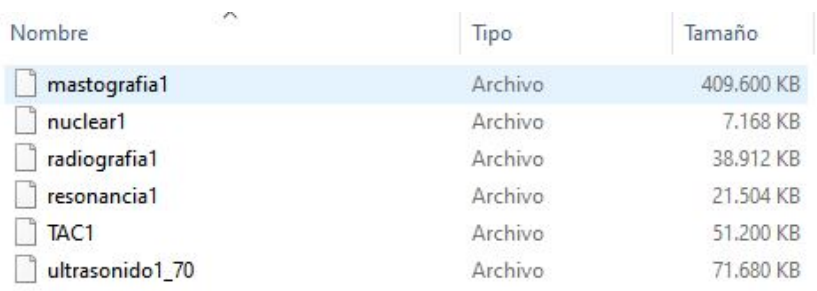



Figura A.8: Imágenes médicas generadas para las pruebas con TCP

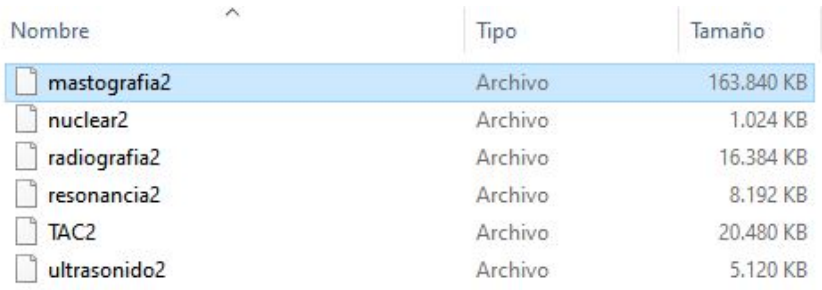



Figura A.9: Imágenes médicas generadas para las pruebas con UDP

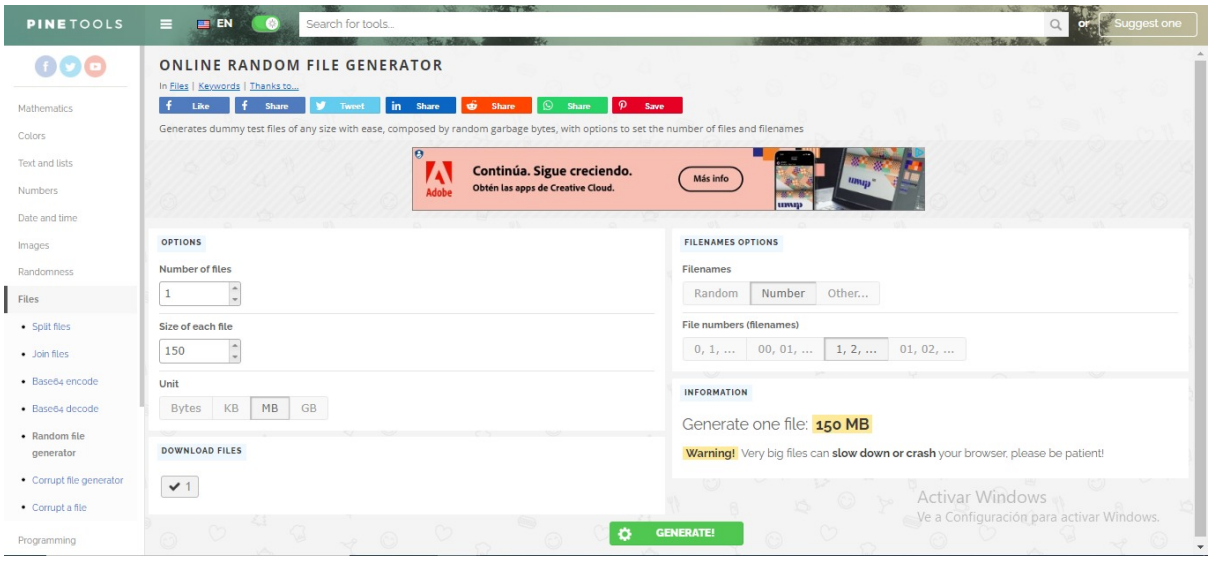



Figura A.10: Generador de archivos random online

```
iperf 3.1.3
CYGWIN_NT-10.0 DESKTOP-AEBUJCB 2.5.1(0.297/5/3) 2016-04-21 22:14
   x86_64
Time: Thu, 13 Jul 2023 19:51:41 GMT
Connecting to host 192.168.48.117, port 5201
      Cookie: DESKTOP-AEBUJCB.1689277901.045439.45
      TCP MSS: 0 (default)
[  4] local 192.168.54.74 port 59230 connected to 192.168.48.117
   port 5201
Starting Test: protocol: TCP, 1 streams, 131072 byte blocks,
   omitting 0 seconds, 200 second test
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec  6.88 MBytes  57.6 Mbits/sec
[  4]   1.00-2.00   sec  7.25 MBytes  60.9 Mbits/sec
[  4]   2.00-3.00   sec  7.50 MBytes  62.9 Mbits/sec
[  4]   3.00-4.00   sec  6.88 MBytes  57.7 Mbits/sec
[  4]   4.00-5.00   sec  6.50 MBytes  54.5 Mbits/sec
[  4]   5.00-6.00   sec  4.50 MBytes  37.7 Mbits/sec
[  4]   6.00-7.00   sec  5.88 MBytes  49.2 Mbits/sec
[  4]   7.00-8.00   sec  6.00 MBytes  50.3 Mbits/sec
[  4]   8.00-9.00   sec  6.88 MBytes  57.8 Mbits/sec
[  4]   9.00-10.00  sec  6.75 MBytes  56.5 Mbits/sec
[  4]  10.00-11.00  sec  6.25 MBytes  52.5 Mbits/sec
[  4]  11.00-12.00  sec  6.00 MBytes  50.3 Mbits/sec
[  4]  12.00-13.00  sec  5.00 MBytes  42.0 Mbits/sec
[  4]  13.00-14.00  sec  6.50 MBytes  54.5 Mbits/sec
[  4]  14.00-15.00  sec  7.00 MBytes  58.7 Mbits/sec
[  4]  15.00-16.00  sec  6.62 MBytes  55.5 Mbits/sec
[  4]  16.00-17.00  sec  6.75 MBytes  56.7 Mbits/sec
[  4]  17.00-18.00  sec  7.00 MBytes  58.7 Mbits/sec
[  4]  18.00-19.00  sec  6.25 MBytes  52.5 Mbits/sec
[  4]  19.00-20.00  sec  5.62 MBytes  47.1 Mbits/sec
[  4]  20.00-21.00  sec  7.00 MBytes  58.8 Mbits/sec
```

```
[  4]  21.00-22.00  sec  5.88 MBytes  49.2 Mbits/sec
[  4]  22.00-23.00  sec  5.88 MBytes  49.2 Mbits/sec
[  4]  23.00-24.00  sec  6.50 MBytes  54.6 Mbits/sec
[  4]  24.00-25.00  sec  6.50 MBytes  54.5 Mbits/sec
[  4]  25.00-26.00  sec  6.12 MBytes  51.3 Mbits/sec
[  4]  26.00-27.00  sec  5.25 MBytes  44.1 Mbits/sec
[  4]  27.00-28.00  sec  6.12 MBytes  51.4 Mbits/sec
[  4]  28.00-28.01  sec   128 KBytes  85.1 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
Test Complete. Summary Results:
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-28.01  sec   177 MBytes  53.1 Mbits/sec
                  sender
      Sent  177 MByte /  400 MByte (44%) of mastografia1
[  4]   0.00-28.01  sec   177 MBytes  53.1 Mbits/sec
                  receiver
CPU Utilization: local/sender 5.5% (0.7%u/4.8%s), remote/receiver
   1.3% (0.4%u/0.8%s)

iperf Done.
```

Listing A.1: Informe de resultados para el archivo **mastografia** utilizando el protocolo TCP

### A.4. Pasos para la instalación de MiniNet y Opendaylight.

1. Desde el terminal de la máquina virtual ejecutar:
   1.1 sudo apt-get update
   1.2 sudo apt-get upgrade
   1.3 git clone https://github.com/mininet/mininet
   1.4 cd mininet
   1.5 git tag git checkout -b mininet-2.3.1b4 2.3.1b4
   1.6 cd ..
   1.7 mininet/util/install.sh -a

```
1.8 sudo mn --switch ovsbr --test pingall
```

Listing A.2: Pasos para instalar MiniNet

1. Desde el terminal de la máquina virtual ejecutar:
   ```
   1.1 sudo apt-get -y update
   1.2 sudo apt-get -y upgrade
   1.3 sudo apt-get -y install unzip
   1.4 sudo apt-get install openjdk-8-jre
   1.5 sudo update-alternatives --config java
   1.6 ls -l /etc/alternatives/java
   1.7 echo 'export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/'jre
       >> ~/.bashrc
   1.8 source ~/.bashrc
   1.9 echo $JAVA_HOME
   1.10 curl -XGET -O https://nexus.opendaylight.org/content/
       repositories/opendaylight.release/org/opendaylight/
       integration/karaf/0.8.0/karaf-0.8.0.zip
   1.11 unzip karaf-0.8.0.zip
   1.12 cd ~/karaf-0.8.0
   1.3 ./bin/karaf
   ```

2. Al ingresar al entorno del controlador por primera vez se deben instalar los siguientes componentes ejecutando los siguientes comandos:
   ```
   2.1 feature:install odl-restconf-all
   2.2 feature:install odl-openflowplugin-app-forwardingrules-
       manager
   2.3 feature:install odl-openflowplugin-app-forwardingrules-sync
   2.4 feature:install odl-openflowplugin-app-southbound-cli
   2.5.feature:install odl-openflowplugin-app-table-miss-enforcer
   2.6 feature:install odl-openflowplugin-app-topology
   2.7 feature:install odl-openflowplugin-flow-services-rest
   2.8 feature:install odl-openflowplugin-libraries
   2.9 feature:install odl-l2switch-all
   ```

```
2.10 feature:install odl-l2switch-switch-ui
2.11 feature:install odl-mdsal-all
2.11 feature:install odl-dlux-core
2.12 feature:install odl-dluxapps-applications
2.13 feature:install odl-dluxapps-yangutils
2.14 feature:install odl-yanglib
2.15 feature:install odl-vtn-manager
2.16 feature:install odl-vtn-manager-rest
2.17 feature:install odl-vtn-manager-neutron
2.18 feature:install odl-config-all
2.19 feature:install odl-netconf-all
2.20 feature:install features-yanglib
2.21 feature:install features-dlux
2.22 feature:install features-dluxapps
2.23 feature:install features-l2switch
2.24 feature:install features-openflowplugin

3. Es posible ingresar desde cualquier navegador mediante la
   siguiente url:
http://[IP del controlador]:8181/index.html
```

Listing A.3: Pasos para instalar OpenDaylight

## A.5. Código de python para la topología y configuración del controlador

```
from mininet.net import Mininet
from mininet.node import Controller, RemoteController, OVSController
from mininet.node import CPULimitedHost, Host, Node
from mininet.node import OVSKernelSwitch, UserSwitch
from mininet.node import IVSSwitch
from mininet.cli import CLI
from mininet.log import setLogLevel, info
from mininet.link import TCLink, Intf
from subprocess import call
import subprocess

def netTAR():
    #Función para limpiar las interfaces de MiniNet
    subprocess.run(['sudo', 'mn', '-c'])
    #Función para conectar a un controlador remoto y los switch tipo
OVSKernelSwitch
    net = Mininet(controller=RemoteController, switch=OVSKernelSwitch)
    #Conectar el controlador remoto con la dirección IP y el puerto
especificado
    c1 = net.addController('c1', controller=RemoteController,
ip="10.22.147.185", port=6653)

    #Hosts de la VLAN 48-Red WiFi
    h11_48 = net.addHost( 'h11_48' , ip='192.168.48.11')
    h12_48 = net.addHost( 'h12_48' , ip='192.168.48.12')
```

Figura A.11: Parte del código de python dónde se diseñó la topología de red. El código completo se encuentra disponible en: https://n9.cl/8lt0g

```
[
    {
        "sentData": {
            "update-vtn": {
                "input": {
                    "tenant-name": "WiFi",
                    "update-mode": "CREATE",
                    "operation": "ADD",
                    "description": "VTN para la red WiFi",
                    "idle-timeout": "0",
                    "hard-timeout": "0"
                }
            }
        },
        "receivedData": {
            "update-vtn": {
                "output": {
                    "status": "CREATED"
                },
                "input": {
                    "tenant-name": "WiFi",
                    "update-mode": "CREATE",
                    "operation": "ADD",
                    "description": "VTN para la red WiFi",
                    "idle-timeout": "0",
                    "hard-timeout": "0"
                }
            }
        },
        "path": "http://10.22.147.185:8181/restconf/operations/vtn:update-vtn",
        "collection": "tesis",
        "method": "POST",
```

Figura A.12: Parte de la configuración en el controlador. El código completo se encuentra disponible en: https://n9.cl/kbbss

## Glosario

**Analizador de red:** software utilizado para capturar, monitorear y analizar el tráfico de red con el fin de diagnosticar problemas, identificar patrones de uso y optimizar el rendimiento de la red.

**Ancho de banda:** medida de la capacidad máxima de una red para transmitir datos, representando la cantidad de información que puede ser transferida en un período de tiempo dado.

**Application Programming Interface (API)**: conjunto de reglas y protocolos que permiten a diferentes aplicaciones de software comunicarse y compartir datos entre sí de manera estandarizada y programática.

**Arquitectura, topología o infraestructura de red:** estructura física que define cómo los dispositivos están interconectados y organizados en una red.

**Backbone:** infraestructura central de alta capacidad que interconecta redes más pequeñas.

**Cable UTP (Unshielded Twisted Pair)**: tipo de cable de red ampliamente utilizado para la transmisión de datos en redes LAN, que consta de pares de hilos de cobre trenzados sin protección externa.

**Calidad de Servicio (QoS):** capacidad de una red para garantizar un nivel deseado de rendimiento y priorización de tráfico, asegurando una distribución eficiente y confiable de los recursos de red.

**Casa de salud:** institución o centro de atención médica donde se brindan servicios de salud a los pacientes.

**Controlador SDN:** componente centralizado en una arquitectura SDN que gestiona y controla el plano de control, tomando decisiones de enrutamiento y configuración de la red.

**Control de redes:** conjunto de técnicas, herramientas y procesos utilizados para gestionar y regular el flujo de tráfico y recursos en una red, asegurando un funcionamiento eficiente y óptimo.

**Digital Imaging and Communications in Medicine (DICOM):** Estándar de imágenes médicas para compartir y almacenar datos clínicos.

**Distribución de flujos:** proceso de direccionamiento y gestión de flujos de datos a través de una red, para garantizar una distribución eficiente y controlada del tráfico de red.

**Escalabilidad:** capacidad de una red para crecer y adaptarse de manera eficiente y flexible a medida que aumenta el número de dispositivos, usuarios o demanda de recursos.

**Gestión de redes:** administración y control de una red, incluyendo tareas como la supervisión, configuración, mantenimiento y solución de problemas para garantizar su rendimiento y

seguridad.

**Imágenes médicas:** representaciones visuales utilizadas en el diagnóstico y tratamiento de enfermedades, obtenidas a través de diferentes modalidades de imagen como radiografías, resonancias magnéticas, tomografías computarizadas, etc.

**Internet Protocol (IP):** protocolo de comunicación utilizado en Internet y redes de computadoras para asignar direcciones únicas a dispositivos conectados a la red y facilitar la transmisión de datos entre ellos.

**Jitter:** variación en los tiempos de llegada de los paquetes de datos, lo que puede resultar en una reproducción irregular o interrupciones en aplicaciones en tiempo real como voz o video.

**Latencia:** tiempo que tarda un paquete de datos en viajar desde su origen hasta su destino en una red, influenciado por la velocidad de transmisión y la distancia recorrida.

**Mastografía:** imagen médica capturada mediante radiación ionizante que muestra las estructuras internas del cuerpo, útil para el diagnóstico y seguimiento de condiciones médicas.

**Medicina nuclear:** método de imagen médica que emplea campos magnéticos y ondas de radio para generar imágenes detalladas de tejidos y órganos internos del cuerpo, sin usar radiación ionizante.

**MiniNet:** herramienta de emulación de redes de código abierto que permite crear una red virtualizada en un único host, facilitando la experimentación y pruebas de prototipos de redes de forma escalable y controlada.

**Modelo Differentiated Services (DiffServ)**: es un enfoque de QoS que busca clasificar y gestionar el tráfico de red en diferentes clases de servicios, asignando prioridades y tratamientos específicos a cada clase para garantizar un rendimiento adecuado y una distribución eficiente de los recursos.

**Niveles de Servicios (SLA):** acuerdo que establece los niveles de servicio acordados entre un proveedor y un cliente.

**OpenDaylight:** plataforma de control de redes de código abierto que proporciona una base flexible y programable para implementar y gestionar redes SDN, facilitando la interoperabilidad y la innovación en entornos de red.

**Picture Archiving and Communication System (PACS)**: sistema de archivo y comunicación de imágenes utilizado en entornos médicos para el almacenamiento, gestión y distribución de imágenes médicas digitales.

**Pérdida de paquetes:** fallo en la entrega exitosa de paquetes de datos en una red.

**Plano de Control:** capa de la arquitectura de red encargada de tomar decisiones y controlar el enrutamiento y las políticas de gestión de la red.

**Plano de Datos:** capa de la arquitectura de red encargada de transmitir los paquetes de datos entre dispositivos, sin realizar decisiones o control de la red.

**Políticas de red:** reglas y directrices establecidas para definir el acceso, uso y administración de los recursos de una red de computadoras.

**Prioridad de servicios:** asignación de niveles de importancia o preferencia a diferentes tipos de tráfico de datos, con el fin de garantizar un tratamiento preferencial y una distribución adecuada de los recursos de red.

**Rack:** estructura para montar dispositivos electrónicos en forma vertical.

**Radiografía:** imagen médica capturada mediante radiación ionizante que muestra las estructuras internas del cuerpo, útil para el diagnóstico y seguimiento de condiciones médicas.

**Red:** interconexión de dispositivos informáticos y sistemas, que permite la comunicación y el intercambio de información entre ellos.

**Redes Definidas por Software (SDN):** enfoque de gestión de redes que separa el plano de control del plano de datos, permitiendo una mayor flexibilidad y programabilidad de las redes.

**Redes dinámicas:** redes en las que los dispositivos pueden unirse o abandonar la red de manera flexible, permitiendo una configuración y reconfiguración automática de la topología de la red para adaptarse a cambios en la infraestructura o la demanda.

**Redes Tradicionales (RT):** infraestructura de redes convencionales que utilizan routers, switches y cables físicos para interconectar dispositivos y permitir la comunicación de datos.

**Rendimiento de la red:** medida de la capacidad y eficiencia de una red para transmitir datos de manera rápida y confiable.

**Resonancia:** método de imagen médica que emplea campos magnéticos y ondas de radio para generar imágenes detalladas de tejidos y órganos internos del cuerpo, sin usar radiación ionizante.

**Salud Digital o telesalud:** aplicación de tecnologías de la información en el ámbito de la salud para mejorar la atención y gestión de los servicios médicos.

**Switch:** dispositivo de red para conectar y gestionar la comunicación entre múltiples dispositivos en una red local.

**Switch OpenFlow (S-OF):** switch compatible con el protocolo OpenFlow utilizado en arquitecturas SDN.

**Tecnologías de la Información y Comunicación (TIC):** conjunto de herramientas, dispositivos y recursos tecnológicos utilizados para procesar, transmitir y almacenar información, facilitando la comunicación y el acceso a la información de manera digital.

**Telediagnóstico:** práctica médica en la que se utilizan las TICs a distancia para realizar diag-

nósticos médicos y emitir recomendaciones sin la presencia física del paciente.

**Telemedicina:** prestación de servicios médicos y de salud a distancia, a través de TICs, permitiendo la consulta, diagnóstico y tratamiento remoto de los pacientes.

**Throughput:** cantidad de datos que puede ser transmitida exitosamente a través de una red en un periodo de tiempo determinado, generalmente expresada en bits por segundo (bps).

**Tomografía axial computarizada (TAC):** técnica de imagen médica que utiliza ondas sonoras de alta frecuencia para generar imágenes en tiempo real de estructuras internas del cuerpo, útil para diagnósticos médicos y seguimiento de condiciones.

**Tráfico de datos:** flujo de información digital que se transmite a través de una red.

**Tráfico TCP:** datos transmitidos a través de la red utilizando el Protocolo de Control de Transmisión (TCP) para una comunicación confiable.

**Ultrasonido:** técnica de imagen médica que utiliza ondas sonoras de alta frecuencia para generar imágenes en tiempo real de estructuras internas del cuerpo, útil para diagnósticos médicos y seguimiento de condiciones.

**Wereables:** dispositivos electrónicos portables que brindan funcionalidades diversas como seguimiento de actividad física y monitoreo de salud.