Mecanismo de control de congestión para la latencia en un network slice dedicado a URLLC: un caso de estudio en cirugía remota

Trabajo de Grado

Kevin Muñoz Rengifo
Juan Manuel Solís Prado

Director: PhD. Oscar Mauricio Caicedo Rendón
Codirector: Msc. Johanna Andrea Hurtado Sanchez

Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2022

Mecanismo de control de congestión para la latencia en un slice dedicado a URLLC: un caso de estudio en cirugía remota

Kevin Muñoz Rengifo
Juan Manuel Solís Prado

Trabajo de grado presentado a la Facultad de Ingeniería
Electrónica y Telecomunicaciones de la
Universidad del Cauca para obtener el título de:
Ingeniero en Electrónica y Telecomunicaciones

Director: PhD. Oscar Mauricio Caicedo Rendón
Codirector: Msc. Johanna Andrea Hurtado Sanchez

Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2022

Agradecimentos

A nuestras familias, especialmente a nuestros padres, madres y hermanos por su apoyo incondicional en todos los momentos vividos de este proceso. Agradecer a todas las personas que hicieron parte de nuestra vida universitaria, incluyendo amigos, compañeros del programa de Ingeniería Electrónica y Telecomunicaciones y especialmente a los integrantes de EOL Solutions, los cuales fueron un pilar y apoyo importante en toda nuestra etapa formativa. Agradecer a nuestro director Oscar Mauricio Caicedo, quien se convirtió en nuestro mentor académico, personal y profesional. Mencionar además, al semillero de investigación COMSOCAUCA y en especial a William Fernando Villota y a Johanna Andrea Hurtado quienes nos brindaron las bases teóricas para la realización de este proyecto de grado. Finalmente, nos sentimos orgullosos por haber realizado nuestra formación académica en la distinguida Universidad del Cauca, donde logramos crear lazos de amistad que perdurarán por toda nuestra vida, conocer profesionales con formaciones y valores humanos excelentes, convirtiéndose así en el lugar donde logramos cumplir metas y sueños, *Posteris lumen moriturus edat.*

Resumen

El objetivo de la tecnología inalámbrica 5G consiste en admitir tres tipos de servicios generales con requisitos muy diferentes: Banda Ancha Móvil Mejorada, Comunicaciones de Tipo de Máquina Masiva y Comunicaciones Ultra Confiables de Baja Latencia o URLLC. En particular, los servicios habilitados para URLLC se caracterizan por los estrictos requisitos de calidad del servicio que exigen, los cuales coexisten dentro de la misma arquitectura de red mediante la segmentación de la misma. Controlar la congestión de la red en los *slices* dedicados a los servicios URLLC es una tarea fundamental para reducir la latencia, el *jitter* y la pérdida de paquetes. En las soluciones más recientes centradas en el control de la congestión en las redes 5G, los autores presentan distintas soluciones basadas en técnicas de aprendizaje automático, tales como el aprendizaje por refuerzo y el aprendizaje profundo. Este trabajo de grado propone un mecanismo de control de congestión para la gestión eficiente de recursos en el núcleo de red 5G orientado a mitigar la latencia en un *slice* de un servicio de URLLC. El mecanismo propuesto es evaluado teniendo en cuenta 3 métricas de desempeño: latencia, tasa de paquetes perdidos, y el *jitter*. Por otra parte, el mecanismo se implementó en un entorno de red emulado utilizando Mininet y Docker, estabilizando la latencia en un 40 % por debajo del umbral establecido de 0.1 milisegundos para el servicio URLLC. Con respecto a la tasa de pérdida de paquetes y el *jitter*, el mecanismo logra estabilizarlos en 0.2 % y 0.01 milisegundos respectivamente, muy lejos del 1 % de tasa de pérdida de paquetes, y 30 milisegundos de *jitter* que se requieren para llevar a cabo con éxito la cirugía remota.

Índice general

Índice de figuras **VII**

Índice de tablas **IX**

Lista de abreviaciones **X**

1. Introducción **1**

1.1. Objetivos 4

1.1.1. Objetivo General 4

1.1.2. Objetivos Específicos 5

1.2. Contribuciones 5

1.3. Organización 6

2. Conceptos Fundamentales y Estado del Arte **7**

2.1. Marco Teórico 7

2.1.1. Redes de Quinta Generación 7

2.1.2. Redes Lógicas 8

2.1.3. Redes Definidas por Software 11
2.1.4. Virtualización de Funciones de Red 12
2.1.5. Cirugía Remota 13
2.1.6. Control de Congestión 13
2.1.7. Aprendizaje por Refuerzo Profundo 15
2.1.7.1. Procesos de Decisión de Markov 15
2.1.7.2. Aprendizaje por Refuerzo 16
2.1.7.3. Aprendizaje Profundo 17
2.1.7.4. Aprendizaje-Q Profundo 19
2.2. Trabajo Relacionado 20
2.2.1. Control de Congestión en 4G 20
2.2.2. Control de Congestión en 5G 24

3. Sistema de Control de Congestión 30

3.1. Descripción General 30
3.2. Arquitectura 31
3.2.1. Gestión y Orquestación de Funciones de Red Virtualizadas . . 32
3.2.1.1. Orquestador de Virtualización de Funciones de Red . 33
3.2.1.2. Gestor de Funciones de Red Virtualizadas 34
3.2.1.3. Gestor de Infraestructura Virtualizada 34
3.2.2. Sistemas Operativos y de Soporte Empresarial 34
3.2.3. Funciones de Red Virtualizadas y Sistema de Gestión de Elementos 35

3.2.4. Infraestructura de Funciones de Red Virtualizadas 36
3.2.5. Mecanismo de Control de Congestión 36
3.3. Agente de Control de Congestión 39
3.3.1. Espacio de Estados 40
3.3.2. Espacio de Acciones 41
3.3.3. Recompensa 41
3.3.4. Redes Neuronales 42
3.3.5. Exploración y Explotación 43
3.4. Algoritmo de Control de Congestión 44

4. Evaluación y Análisis de Resultados 46

4.1. Evaluación 46
4.1.1. Entorno de Evaluación 46
4.1.2. Métricas 48
4.1.3. Experimentación 49
4.2. Resultados y Análisis 53
4.2.1. Latencia 53
4.2.2. Paquetes perdidos 57
4.2.3. *Jitter* 60

5. Conclusiones y Trabajos Futuros 64

5.1. Conclusiones 64
5.2. Trabajos Futuros 65

5.3. Comentarios Finales 65

Bibliografía **65**

Anexos **80**

A. Algoritmos **1**

B. Publicaciones **2**

Índice de figuras

2.1. Redes lógicas 10

2.2. Cirugía remota 13

2.3. Múltiples flujos de tráficos comparten un enlace. 14

2.4. Aprendizaje por refuerzo 16

2.5. Red neuronal artificial 18

2.6. Red-Q profunda 19

3.1. Arquitectura de referencia 32

3.2. Caracterización de grafos para cada tipo de servicio 37

3.3. Arquitectura agente DRL 43

4.1. Topología de evaluación 47

4.2. Arquitectura de ambiente de experimentación 48

4.3. Gráfica de recursos relacionados con la memoria 50

4.4. Recursos asignados y usados con 1 unidad de CPU inicial 51

4.5. Recursos asignados y usados con 3 unidades de CPU iniciales 52

4.6. Recursos asignados y usados con 5 unidades de CPU iniciales 52

4.7. Recursos asignados y usados con 7 unidades de CPU iniciales 53

4.8. Latencia con 1 unidad de CPU inicial 54

4.9. Latencia con 3 unidades de CPU iniciales 54

4.10. Latencia con 5 unidades de CPU iniciales 55

4.11. Latencia con 7 unidades de CPU iniciales 55

4.12. Paquetes perdidos con 1 unidad de CPU inicial 57

4.13. Paquetes perdidos con 3 unidades de CPU iniciales 58

4.14. Paquetes perdidos con 5 unidades de CPU iniciales 58

4.15. Paquetes perdidos con 7 unidades de CPU iniciales 59

4.16. *Jitter* con 1 unidad de CPU inicial 61

4.17. *Jitter* con 3 unidades de CPU iniciales 61

4.18. *Jitter* con 5 unidades de CPU iniciales 62

4.19. *Jitter* con 7 unidades de CPU iniciales 62

Indice de tablas

2.1. Trabajos relacionados en 4G 23

2.2. Trabajos relacionados en 5G 29

3.1. Puntos de referencia 35

4.1. Variación de parámetros de unidades de CPU y número de cirugías remotas 49

4.2. Caracterización de tráfico de una cirugía remota 50

4.3. Resultados de latencia 56

4.4. Resultados paquetes perdidos 60

4.5. Resultados *Jitter* 63

Lista de abreviaciones

4G *Fourth Generation* - Cuarta Generación

5G *Fifth Generation* - Quinta Generación

ACB *Access Class Barring* - Restricción de Clase de Acceso

AMF *Access and Mobility Function* - Función de Gestión de Acceso y Movilidad

ANN *Artificial Neural Network* - Red Neuronal Artificial

AQM *Active Queue Management* - Gestión Activa de Colas

BBR *Bottleneck Bandwidth and Round-trip propagation time* - Ancho de Banda de Cuello de Botella y Tiempo de Propagación de Ida y Vuelta

BDP *Bandwith Delay Product*Producto de Retraso de Ancho de Banda

CCE *Congestion Control Engine* - Motor de Control de Congestión

CDMA *Code Division Multiple Access* - Acceso Múltiple por División de Código

CN *Core Network* - Núcleo de Red

CNN *Convolutional Neural Network* - Red Neuronal Convolucional

CP *Control Plane* - Plano de Control

CPU *Central Processing Unit* - Unidad Central de Procesamiento

CUPS *Control and User Plane Separation* - Separación de Planos de Control y de Usuario

D-ACB *Dynamic Access Class Barring* - Restricción de Clase de Acceso Dinámico

DBE *Drift-based Backlog Estimation* - Estimación de Acumulación Basada en la Deriva

DDQN *Double Deep Q-Network* - Red-Q Profunda Doble

Distributional DQN *Distributional Deep Q-Network* - Red-Q Profunda Distribucional

D-ITG *Distributed Internet Traffic Generator* - Generador de Tráfico de Internet Distribuido

DL *Deep Learning* - Aprendizaje Profundo

DNN *Deep Neural Network* - Red Neuronal Profunda

DRL *Deep Reinforcement Learning* - Aprendizaje por Refuerzo Profundo

DQL *Deep Q-Learning* - Aprendizaje-Q Profundo

DQN *Deep Q-Network* - Red-Q Profunda

Dueling DQN *Dueling Deep Q-Network* - Red-Q Profunda de Duelo

eBCC *enhanced BBR Congestion Control* - Control de Congestión Mejorado Basado en BBR

EECCA *Energy-Efficiency Congestion Control Algorithm* - Algoritmo de Control de Congestión con Eficiencia de Energía

eMBB *enhanced Mobile Broadband* - Banda Ancha Móvil Mejorada

eNB *evolved Node B* - Nodo B evolucionado

EMS *Element Management System* - Sistema de Gestión de Elementos

ENH-SCTP *Enhanced Stream Control Transmission Protocol* - Protocolo de Comunicación de Capa de Transporte Mejorado

EPC *Evolved Packet Core* - Núcleo de Paquetes Evolucionado

ETSI *European Telecommunications Standards Institute* - Instituto Europeo de Normas de Telecomunicaciones

F-DCTCP *Fair-Data Center TCP*TCP Equitativo en Centros de Información

FNN *Feedforward Neural Network* - Red Neuronal Prealimentada

GENI *Global Environment for Network Innovations* - Ambiente Global para Innovaciones de Red

IoT *Internet of Things* - Internet de las Cosas

ITU-R *Radiocommunication Sector of the International Telecommunication Union* - Sector de Radiocomunicaciones de la Unión Internacional de Telecomunicaciones

LTE-A *Long Term Evolution-Advanced* - Evolución a Largo Plazo Avanzado

LP-TCP *Loss-Predictor based TCP*TCP Basado en Predicción de Pérdidas

M2M *Machine To Machine* - Máquina a Máquina

MANO *NFV Management Orchestration* - Gestión y Orquestación NFV

MDP *Markov Decision Process* - Procesos de Decisión de Markov

MEC *Multi-Access Edge Computing* - Computación de Borde de Acceso Múltiple

ML *Machine Learning* - Aprendizaje Automático

MME *Mobility Management Entity* - Entidad de Gestión de Movilidad

mMTC *massive Machine Type Communications* - Comunicaciónes de Tipo Máquina Masiva

NF *Network Function* - Función de Red

NFV *Network Function Virtualization* - Virtualización de Funciones de Red

NFVI *Network Function Virtualization Infrastructure* - Infraestructura de Virtualización de las Funciones de Red

NFVO *Network Function Virtualization Orchestrator* - Orquestador de Virtualización de Funciones de Red

NSL *Network Slicing* - Segmentación de Red

PGW *Packet Gateway* - Puerta de Enlace de Paquetes

Prioritized DQN *Deep Q-Network with Prioritized Experience Replay* - Red-Q Profunda con Reproducción de Experiencia Priorizada

Q-TCP *Q-Learning based TCP* - TCP Basado en Aprendizaje-Q

QCN *Quantized Congestion Notification* - Notificación de Gestión Cuantizada

QoS *Quality of Service* - Calidad del Servicio

RAN *Radio Access Network* - Red de Acceso por Radio

RL *Reinforcement Learning* - Aprendizaje Reforzado

RL-TCP *Reinforcement Learning based TCP* - TCP Basado en Aprendizaje por Refuerzo

RNIS *Radio Network Information Service* - Servicio de Información de Red de Radio

RNN *Recurrent Neural Network* - Red Neuronal Recurrente

SAFE-TS *Self-Adaptive Flexible TTI Scheduling* - Estrategia de Programación de TTI Autoadaptable

SBA *Service-Based Architecture* - Arquitectura basada en Servicios

S-TCP *Scalable Transfer Control Protocol* - Protocolo de Control de Transferencia Escalable

SDN *Software Defined Networking* - Redes Definidas por Software

SMF *Session Management Function* - Función de Gestión de Sesión

SVM *Support Vector Machine* - Máquinas de Vectores de Soporte

TCP *Transmission Control Protocol* - Protocolo de Control de Transmisión

TI *Tactile Internet* - Internet Táctil

TTI *Transmission Time Interval* - Intervalo de Tiempo de Transmisión

URLLC *Ultra Reliable Low Latency Communications* - Comunicaciones Ultra Confiables de Baja Latencia

UP *User Plane* - Plano de Usuario

UPF *User Plane Function* - Función de Plano de Usuario

VIM *Virtualized Infrastructure Manage* - Gestor de Infraestructura Virtualizada

VNF *Virtual Network Function* - Función de Red Virtualizada

VNFM *Virtual Network Function Manager* - Gestor de Funciones de Red Virtualizadas

Capítulo 1

Introducción

Internet Táctil (TI, *Tactile Internet*) se describe como una red de internet que combina una latencia ultra baja con una disponibilidad, fiabilidad y seguridad extremadamente altas [1] [2] [3]. TI habilita diferentes casos de uso en campos como el transporte [4], la industria, [5] y especialmente en la medicina [1]. En este último campo, la cirugía remota es un caso de uso de interés ya que al aislar al cirujano del quirófano posibilita el acceso a pacientes ubicados en zonas remotas [6] [7]. La cirugía remota consiste en un cirujano que manipula un sistema telequirúrgico robótico que envía señales hápticas (i.e., retroalimentación relacionada con el tacto). El cirujano debe tener en cuenta tanto la retroalimentación visual, como la transmisión de vídeo e imágenes de alta calidad en tiempo real. La latencia es un factor crucial durante el procedimiento quirúrgico; incluso un retraso de comunicación extremadamente bajo puede causar complicaciones quirúrgicas y provocar la muerte del paciente. En general, el rendimiento de la cirugía remota depende de la latencia, la fluctuación y la pérdida de paquetes [8]. Por lo tanto, es un desafío proporcionar una solución que cumpla con estas métricas de rendimiento [9].

Los sistemas de comunicaciones móviles 4G no cumplen con los requisitos de rendimiento necesarios para las aplicaciones del cuidado de la salud basadas en TI [10] [11]. Esto se debe a que los requisitos de baja latencia para aplicaciones críticas no son la preocupación de las redes 4G, las cuales se centran en altas tasas de datos y cobertura [12]. Por esto, se espera que los sistemas de comunicaciones móviles de

próxima generación como las redes 5G ayuden a cumplir con los requisitos de latencia máxima de 10 milisegundos extremo a extremo [1], de los cuales 0.1 milisegundos deben ser satisfechos por el Núcleo de Red 5G (5G CN, 5G *Core Network*) para los casos de uso que habilita TI [13] [14]. Además, la flexibilidad en el diseño de una red 5G es crucial para cumplir con las métricas de rendimiento esperadas de TI en términos de latencia, *jitter* y pérdida de paquetes. Esta flexibilidad consiste en la capacidad de aceptar dinámicamente nuevas solicitudes, que pueden ser cambios en los requisitos de Calidad del Servicio (QoS, *Quality of Service*) [15]. Una forma de lograr tal flexibilidad es creando *slices* [16] [17] [18].

Un *slice* comprende un grupo de Funciones de Red (NF, *Network Function*), recursos y enlaces [19]. La Segmentación de Red (NSL, *Network Slicing*) es entonces la capacidad de personalizar un conjunto de NF para optimizar el uso de la red para cada servicio [17]. Por lo tanto, NSL hace posible que los elementos de la red 5G y sus NF se configuren y reutilicen fácilmente para cumplir con los requisitos específicos del escenario de cirugía remota. La implementación de NSL está destinada a ser una característica de un extremo a otro que incluye el Núcleo de Red (CN, *Core Network*) y la Red de Acceso por Radio (RAN, *Radio Access Network*) de 5G [17]. Por lo tanto, con el NSL, es posible dedicar los recursos de un *slice* específicamente para un escenario de cirugía remota e implementarlo como un servicio [19].

En el 5G CN, las Redes Definidas por Software (SDN, *Software Defined Networking*) (paradigma que rompe la integración vertical al separar la lógica de control de la red (plano de control) de los enrutadores y conmutadores subyacentes que reenvían el tráfico (plano de datos) [20] [21]) y la Virtualización de Funciones de Red (NFV, *Network Function Virtualization*) son candidatos para cumplir con los requisitos de QoS en un *slice* dedicado a habilitar un escenario de cirugía remota (latencia, fluctuación y tasa de pérdida de paquetes) [8] [17] [22]. Sin embargo, según los autores [23] [24] [25], la congestión del tráfico en una red 5G dificulta el cumplimiento de estos requisitos de QoS, especialmente de la latencia [26].

La congestión en la red es uno de los factores críticos para la prestación de servicios de Comunicaciones Ultra Confiables de Baja Latencia (URLLC, *Ultra Reliable Low Latency Communications*) [23] [24] [25]. Esta congestión puede originarse debido a

múltiples solicitudes de tráfico que requieren los *slices* [27] [8]. Particularmente, en el caso de uso de la cirugía remota, se consideran flujos de tráfico pesado (también conocidos como flujos elefantes) como la transmisión de vídeo y de imágenes de alta calidad en tiempo real, y del otro lado, se tienen flujos de tráfico pequeños (también conocidos como flujos ratones), como lo son la transmisión de las señales hápticas. Siendo los flujos elefantes los causantes de la congestión de la red [28] [29] [30] debido a que aumentan el tiempo de procesamiento y por ello la latencia, afectando el tiempo de respuesta crítico definido para este tipo de servicio. Por lo tanto, representa un desafío diseñar una solución de comunicación confiable, segura y de baja latencia, que lidie con el problema de la congestión en la red y que satisfaga los requisitos de calidad del servicio necesarios para este tipo de servicio [9].

En 4G, se han utilizado técnicas de Aprendizaje Automático (ML, *Machine Learning*) para abordar el problema de congestión de la red. En la literatura, [31] utiliza algoritmos de Aprendizaje Profundo (DL, *Deep Learning*) para enrutar el tráfico de red, evitando la congestión de manera eficiente. Los autores de [32] y [33] proponen un enfoque con algoritmos de ML fundamentados en la teoría del aprendizaje estadístico, basándose en el Protocolo de Control de Transmisión (TCP, *Transmission Control Protocol*) y su ventana deslizante de congestión, los cuales, pueden aprender del entorno de red para aliviar la congestión mejorando algunas variantes de TCP como TCP *Tahoe*, TCP *Reno* o TCP *Westwood* [34]. [35] propone un mecanismo de control de congestión basado en SDN para lidiar con la congestión causada por la variante de TCP, TCP *Incast*. Este problema aumenta los tiempos de espera de TCP, lo que también puede dañar el rendimiento de las aplicaciones de centros de datos sensibles a la latencia [36].

Los trabajos [37] y [38] proponen soluciones basadas en ML para manejar la congestión del tráfico de red en 5G. [37] utiliza un algoritmo de árbol de decisión para modelar el detector de anomalías de QoS en la red simulada. Sin embargo, el aprendizaje supervisado (i.e., basado en comportamientos o características analizadas en datos históricos etiquetados [39]) suele ser costoso, es decir, requiere una gran cantidad de datos de entrenamiento a gran escala para redes 5G heterogéneas. [38] usa el Aprendizaje por Refuerzo (RL, *Reinforcement Learning*) combinado con técnicas de DL, es decir, mediante el Aprendizaje por Refuerzo Profundo (DRL, *Deep Rein-*

forcement Learning), donde el control de la congestión se maneja como un problema de toma de decisiones secuencial bajo el marco de RL. Aun así, existen lagunas en el desempeño del mecanismo presentado en [38], debido a que el procesamiento real de paquetes no se tiene en cuenta en sus evaluaciones.

[40], [41] y [42] proponen mecanismos de control de congestión para detectar dónde se produce la congestión mediante técnicas RL/DRL. Los artículos recientes que proponen soluciones basadas en algoritmos DRL muestran un rendimiento superior a las variantes TCP en términos de rendimiento de la red en general. Sin embargo, estos poseen problemas en el mecanismo de DRL para manejar la congestión del *slice* debido a la falta de datos realistas. Además, en la literatura hay muy pocas implementaciones de control de congestión presentes en un *slice* dedicado a un servicio de URLLC con el tráfico variable.

Teniendo en cuenta que el control de la congestión en un *slice* es fundamental para cumplir con los requisitos de QoS, este trabajo de grado destaca la importancia del manejo eficiente de los recursos en el 5G CN puesto que ayuda a mitigar la congestión de tráfico, garantizando el cumplimiento de los requisitos de QoS del caso de uso de cirugía remota. Por lo tanto, este trabajo de grado se centra en resolver la siguiente pregunta de investigación:

¿Cómo abordar la congestión de la red en un *slice* de cirugía remota en el núcleo de red 5G para satisfacer el requisito de latencia?

Para dar respuesta a la pregunta de investigación planteada, presentamos los siguientes objetivos.

1.1. Objetivos

1.1.1. Objetivo General

- Introducir un mecanismo de control de congestión para cumplir con los requisitos de latencia en un *slice* de cirugía remota en el 5G CN.

1.1.2. Objetivos Específicos

- Diseñar un mecanismo de control de congestión para un *slice* de cirugía remota en el 5G CN.
- Desarrollar una implementación de referencia del mecanismo propuesto.
- Evaluar la implementación de referencia en un entorno emulado, en cuanto a métricas de rendimiento.

1.2. Contribuciones

Este trabajo de grado tiene como objetivo lograr los siguientes aportes:

- Un mecanismo de control de congestión para satisfacer el requisito de latencia en un *slice* de cirugía remota en el 5G *Core Network*.
- Implementación de referencia del mecanismo propuesto.
- Evaluación del prototipo en un entorno emulado.

El trabajo presentado en esta monografía fue preparado para enviar a la comunidad científica través de un artículo a una revista indexada (ver Anexo B).

- **Kevin Muñoz Rengifo**, **Juan Manuel Solis Prado**, Oscar Mauricio Caicedo Rendón, Johanna Andrea Hurtado Sanchez. **Congestion Control Mechanism for latency in a 5G Network Slice dedicated to URLLC: a case study in remote surgery**. IEEE Latincom.
  - Estado: Escrito y por enviar
  - Clasificación: B

1.3. Organización

Este documento de trabajo de grado se ha dividido en los capítulos que se describen a continuación.

- El Capítulo 1 presenta la **Introducción** que incluye la declaración del problema, objetivos, contribuciones, y la organización de este documento.
- El Capítulo 2 presenta el **Estado del Arte**, organizado por el **Marco Teórico** sobre los temas relacionados con la investigación realizada (que incluye Redes de Quinta Generación, Redes Lógicas, Cirugía Remota, Control de Congestión y DRL). Incluye además la sección de **Trabajo Relacionado** que describe los trabajos de investigación cercanos a los enfoques propuestos.
- El Capítulo 3 introduce el **Mecanismo de Control de Congestión basado en DRL en el 5G CN**, presentando la **Arquitectura**, seguido del **Agente** y el **Algoritmo** que describe el mecanismo en cuestión.
- El Capítulo 4 presenta la **Evaluación** del mecanismo de control de congestión en varios escenarios de demanda, seguido del **Análisis de Resultados**
- El Capítulo 5 presenta las **Conclusiones** obtenidas, los **Trabajos Futuros** y **Comentarios Finales**.

Capítulo 2

Conceptos Fundamentales y Estado del Arte

En este capítulo, se presenta el Marco Teórico relacionado con los enfoques para satisfacer los requisitos de QoS en el 5G CN. Primero, se describe 5G, Redes Lógicas, Cirugía Remota, Control de Congestión y DRL. En segundo lugar, se revisa el trabajo relacionado de acuerdo con los enfoques de control de congestión existentes.

2.1. Marco Teórico

2.1.1. Redes de Quinta Generación

5G es la quinta generación de las tecnologías y estándares de comunicación inalámbrica. 5G ofrece velocidades de datos máximas de varios Gbps, latencia ultrabaja, más confiabilidad, mayor disponibilidad, y un uso más eficiente del espectro en comparación a su predecesor, 4G. Un mayor rendimiento y una mayor eficiencia potencian nuevas experiencias de usuario y conectan nuevas industrias [43].

El Sector de Radiocomunicaciones de la Unión Internacional de Telecomunicaciones (ITU-R, *Radiocommunication Sector of the International Telecommunication*

Union) ha definido tres categorías de casos de uso potenciales, Banda Ancha Móvil Mejorada (eMBB, *enhanced Mobile Broadband*), Comunicaciónes de Tipo Máquina Masiva (mMTC, *massive Machine Type Communications*), y URLLC de la siguiente manera [44]:

- **eMBB:** enfocada en aplicaciones como transmisión de vídeo de alta definición y conexión de eventos a gran escala, caracterizada por el uso de un ancho de banda móvil extenso, con altas velocidades de datos y una mayor cobertura de usuarios.
- **URLLC:** contempla aplicaciones como controles inalámbricos industriales de fabricación, cirugías remotas, automatización de redes inteligentes y vehículos autónomos, los cuales presentan estrictos requerimientos de confiabilidad, latencia y disponibilidad en la red.
- **mMTC:** comprende aplicaciones como ciudades inteligentes, redes de sensores hospitalarios y sector agrícola inteligente, los cuales se identifican por conectar una gran y diversa cantidad de dispositivos de Internet de las Cosas (IoT, *Internet of Things*) de bajo tráfico de datos sin sensibilidad al retraso.

Diseñar una red que pueda admitir simultáneamente una amplia variedad de casos de uso y requisitos exigentes de QoS, como los de la cirugía remota, todo con un solo conjunto de NF estándar, sería extraordinariamente complicado y costoso. Una alternativa es el NSL, vital para cumplir con los diversos requisitos para las redes 5G, incluida la escalabilidad y la flexibilidad de la red [45].

2.1.2. Redes Lógicas

5G permite la coexistencia de servicios heterogéneos dentro de la misma arquitectura de red por medio del NSL [46]. El *slice* asigna a la red recursos de computación, de almacenamiento y de comunicación entre los servicios activos con el fin de garantizar su aislamiento y unos niveles de rendimiento óptimos [47]. Se conoce como NSL el dividir la red física en redes lógicas separadas llamadas *slices*. Cada *slice* puede ser

configurado para ofrecer capacidades de red y características de red específicas. Un *slice* extremo a extremo puede ayudar a implementar varios servicios basados en 5G, como eMBB, URLLC, y mMTC.

El concepto de NSL fue propuesto por primera vez por la iniciativa Ambiente Global para Innovaciones de Red (GENI, *Global Environment for Network Innovations*), esencialmente creando *slices* separados de un extremo al otro [48]. Cada *slice* consta de una topología de red dedicada, nodos virtuales y protocolos [48]. Este concepto de NSL consta de tres capas: la capa de instancia de servicio, la capa de instancia de NSL y la capa de recursos. La capa de instancia de servicio representa los servicios que deben ser compatibles, por lo que cada uno está representado por una instancia de servicio. La capa de instancia de NSL proporciona las características de red que requiere una instancia de servicio. Finalmente, la capa de recursos son todos aquellos componentes físicos y lógicos que posee la red [49].

NSL permite la creación de valor para *slices* verticales, proveedores de aplicaciones y terceros que carecen de infraestructura de red, ofreciendo radio, redes y recursos en la nube, lo que permite una operación de red personalizada y una verdadera diferenciación del servicio. El NSL debe implementarse de un extremo a otro para satisfacer diversas necesidades comerciales, es por esto que cada *slice* puede tener su propia arquitectura y protocolos de red. NSL implica dividir la RAN, el 5G CN e incluso los host de usuario final. La división de la RAN se puede lograr a través de la abstracción lógica de los recursos de radio (como el espectro) y del hardware físico (como las estaciones base). SDN y NFV permiten configurar los recursos de la red virtualmente de manera flexible, como el ancho de banda de la red, la capacidad de procesamiento del servidor y la capacidad de procesamiento de los elementos de la red. Además, un *slice* se puede construir en el 5G CN para satisfacer necesidades comerciales específicas,[50]

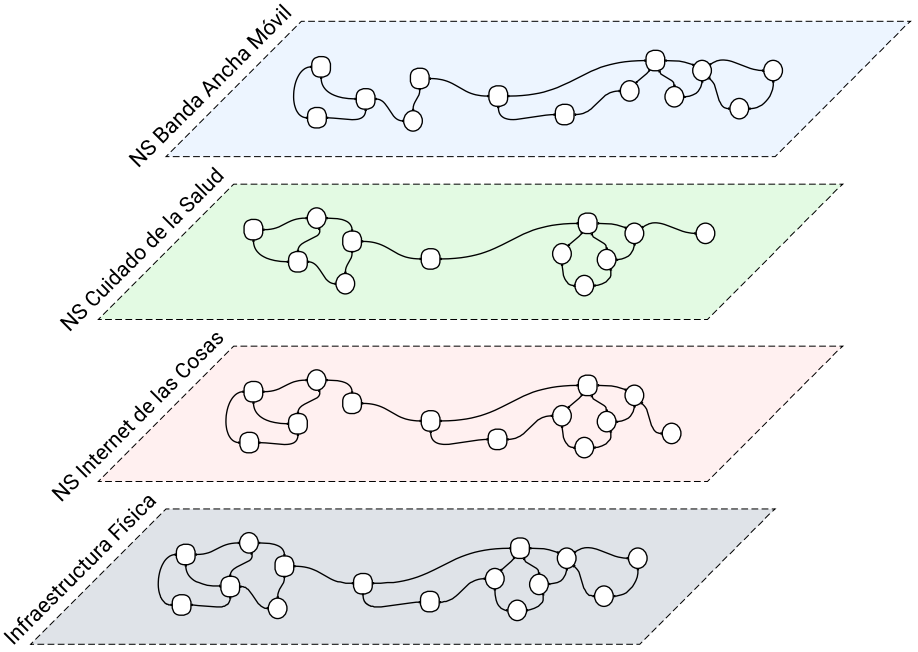

Figura 2.1: Redes lógicas

Como se muestra en la Figura 2.1, un *slice* comprende una colección de recursos (es decir, una unidad maleable, definida por un conjunto de atributos o capacidades), que cumplen con los requisitos de QoS, que pueden variar según el campo de aplicación. En el NSL, se consideran dos tipos de recursos [51], las NF y los recursos de infraestructura. Las NF como bloques funcionales que brindan capacidades de red específicas para soportar y realizar los servicios que pueden ser físicos o virtualizados. Y por otra parte, los recursos de infraestructura, es decir, hardware y software necesarios para alojar y conectar una o más NF. Cada *slice* se asigna a uno o más servicios para minimizar el costo operativo; por lo tanto, NSL es una solución ideal para administrar diferentes sectores y servicios de manera independiente, teniendo especial cuidado con la congestión generada por el tráfico creado como resultado de estos servicios [51] [52].

2.1.3. Redes Definidas por Software

SDN permite un plano de control centralizado y programable, además de brindar abstracción de plano de datos. En SDN los planos de control y de datos están separados, de modo que los operadores de red pueden controlar y administrar directamente sus propios recursos y redes virtualizadas [53].

Las razones por las que SDN es necesario para los operadores de red de servicios se resume a continuación:

- **El aumento de la inteligencia de red y el uso compartido de recursos en las redes de los operadores**: gran parte de la información se almacena y procesa en computadoras en las redes de los operadores (por ejemplo, centros de datos, nubes, etc.). Por lo tanto, los operadores de red necesitan usar tecnologías SDN para controlar y administrar de manera más fácil y eficiente la inteligencia y los recursos en sus redes [53].
- **La necesidad de capacidad de programación de red**: los operadores de red necesitan sistemas de control más inteligentes para orquestar directamente el comportamiento de miles de enrutadores y conmutadores [53].
- **Creación de redes conscientes de los servicios emergentes en las redes**: dado que los clientes desean establecer sus servicios de propósito específico, los operadores de red deben proporcionar un método determinado para la interacción entre los servicios y la infraestructura de la red. Luego los servicios deben aislarse de forma segura del tráfico de otros clientes existentes [53].
- **Creación de redes conscientes de los servicios emergentes en las redes**: a medida que aumenta la complejidad de las redes, los operadores de redes desean reducir la complejidad de la gestión y las operaciones con SDN en lugar de redes configuradas [53].

2.1.4. Virtualización de Funciones de Red

La virtualización de las funciones de red NFV es un enfoque de red propuesto por El Instituto Europeo de Normas de Telecomunicaciones (ETSI, *European Telecommunications Standards Institute*) que permite la sustitución de dispositivos hardware, tales como routers, firewalls y balanceadores de carga, por dispositivos basados en software que se ejecutan como máquinas virtuales en servidores estándar. NFV desacopla las funciones de red de los dispositivos hardware y las traslada a uno o varios servidores virtuales, que pueden cumplir múltiples funciones en un único servidor físico. Este enfoque reduce los costos y minimiza el mantenimiento, debido a que los dispositivos virtuales reemplazan dispositivos de red basados en hardware dedicado [54].

NFV promete a los proveedores de servicios de telecomunicaciones más flexibilidad para extender su capacidad de red y servicios a los usuarios, junto con la capacidad de implementar nuevos servicios de red más rápidos y baratos para brindar mayor agilidad. En resumen, las principales características que ofrece NFV son las siguientes:

Desacoplamiento del software del hardware: Como el elemento de red ya no es una composición de hardware integrada y entidades de software, la evolución de ambas es independiente el uno del otro [55].

Despliegue flexible de funciones de red: La separación del software y el hardware ayuda a reasignar y compartir los recursos de la infraestructura. De esta manera, hardware y software pueden realizar diferentes funciones en distintos momentos. Esto ayuda a que los operadores de red implementen nuevos servicios de red más rápido en la misma plataforma física. Por lo tanto, los componentes pueden ser instanciados en cualquier dispositivo habilitado para NFV en la red y sus conexiones se pueden configurar de forma flexible [55].

Escalado dinámico: El desacoplamiento de la función de red en componentes software instanciables, proporciona una mayor flexibilidad para escalar el rendimiento real de una *Virtual Network Function* (VNF) de una manera más dinámica y con una granularidad más fina [55].

2.1.5. Cirugía Remota

En el caso de uso de URLLC, el servicio de cirugía remota apoya la realización de un procedimiento médico donde el cirujano y el paciente están separados geográficamente. Este servicio involucra un dominio maestro y un dominio esclavo conectado a través de una red de comunicación altamente confiable [9]. El dominio maestro comprende al cirujano y la interfaz humano-sistema necesaria para controlar el dominio esclavo, compuesto como mínimo de brazos robóticos, sensores hápticos, cámara de alta definición, micrófono, y un equipo de asistencia. Los elementos hápticos incluyen sensores de fuerza para mejorar la destreza y precisión de instrumentos quirúrgicos y sensores táctiles. Los enlaces de circuito cerrado están compuestos de enlaces de reenvío y retroalimentación que comunican los dominios maestro y esclavo [56] [57] [58] [59] (ver Figura 2.2).

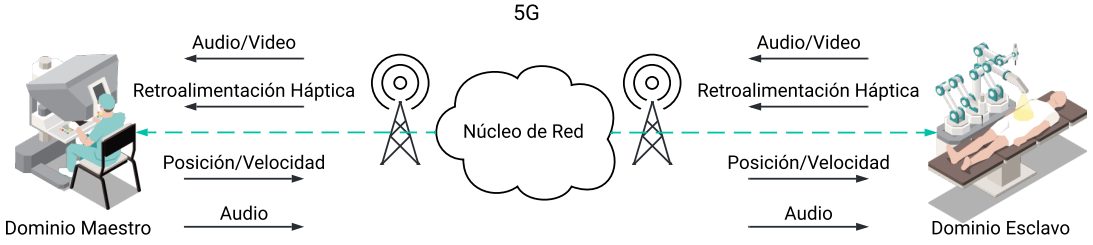

Figura 2.2: Cirugía remota

Los enlaces directos transportan los datos hápticos generados por el cirujano, incluidas las señales hápticas y de voz. Los enlaces de retroalimentación transportan datos sensoriales multi-modales como video, voz, retroalimentación forzada, retroalimentación táctil y los datos fisiológicos del paciente desde el dominio esclavo al dominio maestro [60].

2.1.6. Control de Congestión

En las redes hoy por hoy múltiples usuarios compiten por los escasos recursos que esta posee. En consecuencia, para utilizar los recursos de la red y brindar una bue-

na experiencia de usuario de manera eficiente, las tasas de transmisión de datos de diferentes fuentes de tráfico deben modularse dinámicamente. Este concepto se denomina control de congestión, que es fundamental para la investigación y la práctica de las redes de información [38]. De hecho, según varios autores [23] [24] [25], la congestión de la red dificulta el cumplimiento de los requisitos de QoS, lo que tiene un impacto negativo en la experiencia del usuario.

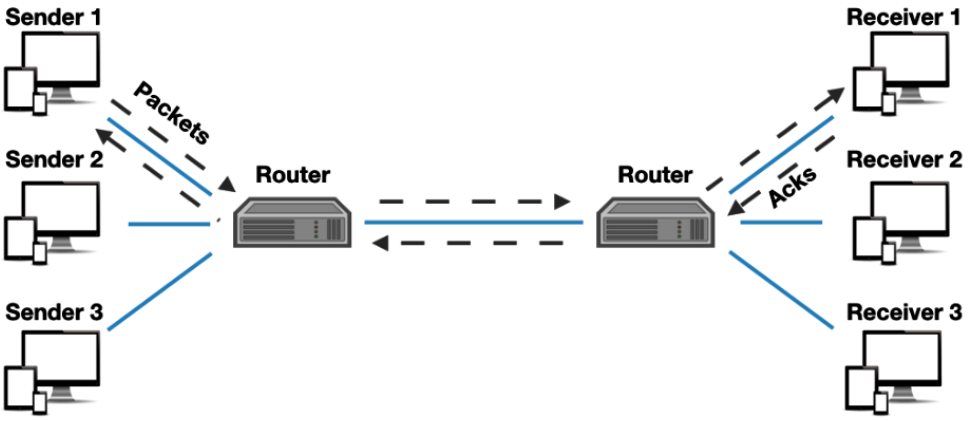

Figura 2.3: Múltiples flujos de tráficos comparten un enlace [38].

La figura 2.3 muestra varias conexiones (también conocidas como flujos) que comparten un único enlace de comunicación. Cada conexión consta de un emisor de tráfico y un receptor de tráfico. El emisor transmite paquetes de datos al receptor y regularmente recibe retroalimentación sobre los paquetes enviados en forma de recibido de paquetes. El emisor ajusta su tasa de transmisión en respuesta a esta retroalimentación. La forma en que se ajusta la tasa de envío está determinada por el protocolo de control de congestión empleado por los dos puntos finales [38]. La interacción de las conexiones de punto final da lugar a dinámicas de red, como la capacidad de los enlaces (ancho de banda), el tamaño del búfer de los enlaces y la política de colas de paquetes, que determina cómo (y de quién) se descarta el exceso de tráfico.

Los campos emergentes de aplicaciones de TI habilitados para 5G están progresando hacia una interacción precisa de persona a máquina y de máquina a máquina, con ejemplos clave que se encuentran en la industria, la robótica, el tráfico vial y la

atención médica [3]. En este último campo, la cirugía remota es una de las aplicaciones destacadas [61], exigiendo una baja latencia (menor de 10 milisegundos) para que la aplicación sea segura [1]. Por lo tanto, el control de la congestión surge como una forma de satisfacer los requisitos de latencia en la red 5G [62], por otra parte, investigaciones recientes muestran que con ayuda de algoritmos basados en ML este control de congestión se hace mas eficiente y efectivo.

2.1.7. Aprendizaje por Refuerzo Profundo

DRL como concepto engloba temáticas que han evolucionado con los años, empezando con los Procesos de Decisión de Markov (MDP, *Markov Decision Process*), seguido de RL y de DL. DL en DRL se utiliza para mejorar la eficiencia y el rendimiento en términos de aprendizaje de RL.

2.1.7.1. Procesos de Decisión de Markov

MDP es un proceso de control estocástico en tiempo discreto. MDP proporciona un marco matemático para modelar problemas de toma de decisiones en los que los resultados son en parte de carácter aleatorio y están bajo el control de la toma de decisiones o un agente. Los MDP son utilizados en el estudio de problemas de optimización que pueden ser resueltos mediante técnicas de RL. MDP se define mediante una tupla (S, A, p, r) donde S es un conjunto finito de estados, A es un conjunto finito de acciones, p es una probabilidad de transición del estado s al estado s' después de que la acción a es ejecutada y finalmente r es la recompensa que se obtiene después de realizar la acción a. El objetivo de un MDP es encontrar una política óptima para maximizar la recompensa [63]. En los MDP se encuentran principalmente dos variantes:

- **Procesos de decisión de Markov parcialmente observables:** MDP asume que el estado del sistema es totalmente observable por el agente. No obstante, en algunos casos el agente soló observa una parte del conjunto de estados del sistema por lo que su política se basa en el descubrimiento a raíz de la elección

de acciones óptimas. Luego, los Procesos de decisión de Markov parcialmente observables se usan para modelar los problemas de toma de decisiones [64].

- **Juegos de Markov:** En teoría de juegos, un juego de Markov, o un juego estocástico, es un juego dinámico con transiciones probabilísticas jugado por múltiples jugadores, en este caso agentes. En un juego de Markov, los agentes comienzan en algún estado inicial, después de observar el estado actual, todos los agentes seleccionan al mismo tiempo sus acciones, recibiendo así las recompensas junto con las nuevas observaciones propias y finalmente transicionando a un nuevo estado. Este proceso se repite en el nuevo estado durante un número finito o infinito de veces. En el juego de Markov, todos los agentes intentan encontrar sus políticas óptimas para maximizar la recompensa [65].

2.1.7.2. Aprendizaje por Refuerzo

RL es una herramienta eficaz y ampliamente utilizada en la literatura por abordar los MDP. En el contexto de RL un agente puede aprender la política óptima a través de la interacción con el entorno. En particular, el agente primero observa su estado actual, luego realiza una acción determinada y recibe una recompensa con un nuevo estado como consecuencia de dicha acción (Ver Figura 2.4) [66].

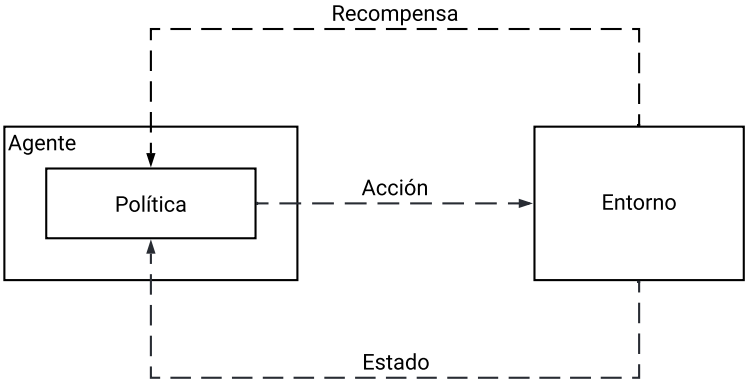

Figura 2.4: Aprendizaje por refuerzo

La información obtenida, es decir, la recompensa y el nuevo estado, es utilizado para ajustar la política del agente, repitiéndose este proceso hasta que la política

del agente se acerque a la política óptima. RL se implementa a través de algoritmos como: Q-learning, Monte Carlo, Dyna-Q o Fuerza Bruta [57], siendo el primero el más efectivo y ampliamente utilizado en la literatura. Finalmente, RL ha permitido contribuir en soluciones a problemas en las redes, donde se presentan aportes en enrutamiento [67], programación de recursos en centros de datos [68] y seguridad informática [69].

2.1.7.3. Aprendizaje Profundo

DL se compone de un conjunto de algoritmos y técnicas útiles para encontrar características importantes de los datos y modelar sus abstracciones de alto nivel. El objetivo principal de DL es evitar la descripción manual de una estructura de datos (como características escritas a mano) mediante el aprendizaje automático de los datos. Por lo general, cualquier red neuronal con dos o más capas ocultas se denomina una Red Neuronal Profunda (DNN, *Deep Neural Network*). La mayoría de los modelos de DL se basan en una Red Neuronal Artificial (ANN, *Artificial Neural Network*). Una ANN es un modelo computacional no lineal basado en la estructura neuronal del cerebro que puede aprender a realizar tareas como clasificación, predicción, toma de decisiones y visualización. Una ANN consta de neuronas artificiales y está organizada en tres capas interconectadas: entrada, oculta y salida (Ver Figura 2.5) [70].

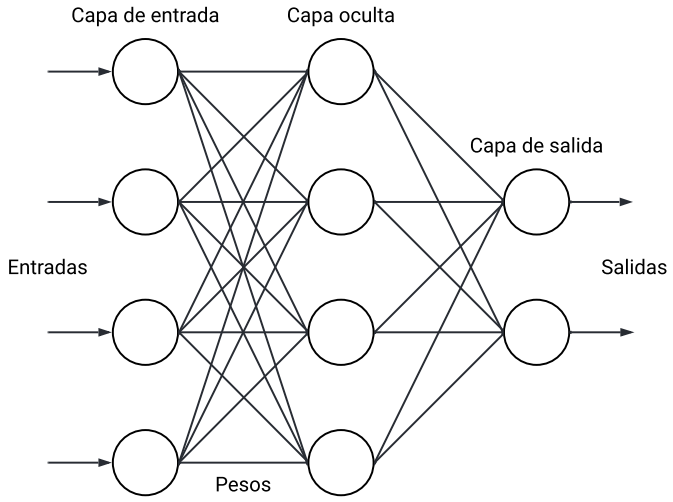

Figura 2.5: Red neuronal artificial

La capa de entrada contiene neuronas de entrada que envían la información a la capa oculta y esta a su ves envía datos a la cada de salida. Cada neurona tiene entradas ponderadas (Sinapsis), una función de activación y una salida. Las sinapsis son los parámetros ajustables que convierten una red neuronal en un sistema parametrizado. La función de activación de un nodo define las salidas de ese nodo dadas las entradas. En particular, la función de activación mapeará los valores de entrada en rangos objetivo según la función de activación seleccionada [70]. Una DNN se define como una ANN con múltiples capas ocultas y en general existen dos modelos DNN estándar:

- **Red Neuronal Prealimentada:** La Red Neuronal Prealimentada (FNN, *Feedforward Neural Network*) se caracteriza en que la información se mueve en una sola dirección, es decir, desde los nodos de la capa de entrada, a través de la capa oculta y hacia los nodos de la capa de salida. En las FNN, la Red Neuronal Convolucional (CNN, *Convolutional Neural Network*) es el modelo más usado en la literatura, especialmente en el reconocimiento de imágenes y voz. La CNN se caracteriza por contener una o más capas convolucionales totalmente conectadas [71].

- **Red Neuronal Recurrente:** La Red Neuronal Recurrente (RNN, *Recurrent Neural Network*) consiste en una variante de una red neuronal artificial recursiva en la que las conexiones entre neuronas forman ciclos, es decir, la salida depende no solo de sus entradas inmediatas, sino también del estado de la neurona de la capa anterior. Esto hace que las RNN sean ideales para aplicaciones con un componente de tiempo [72], por ejemplo, el enrutamiento en redes informáticas [73] y los juegos de vídeo [74].

2.1.7.4. Aprendizaje-Q Profundo

El algoritmo Q-learning de RL puede obtener de manera eficiente una política óptima cuando el espacio de estado y el espacio de acción son pequeños. Sin embargo, en la práctica, con modelos de sistemas complicados, estos espacios suelen ser grandes. Como resultado, es posible que el algoritmo Q-learning no pueda encontrar la política óptima. Por lo tanto, se introduce el Aprendizaje-Q Profundo (DQL, *Deep Q-Learning*) para superar esta deficiencia, este implementa una Red-Q Profunda (DQN, *Deep Q-Network*), es decir, una DNN en lugar de la tabla Q como se muestra en la Figura 2.6 [75].

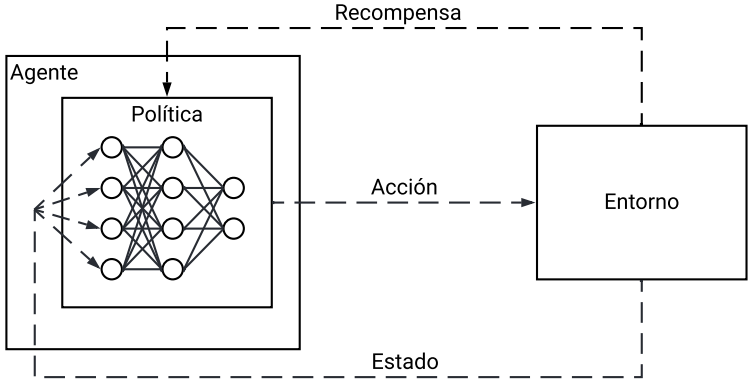

Figura 2.6: Red-Q profunda

DQL hereda y asimila las ventajas de las técnicas de RL y de DL, por lo que, tiene una amplia gama de aplicaciones en la práctica, como el desarrollo de juegos, el transporte y la robótica [76] [77]. Por otra parte, las investigaciones recientes

indican que el uso de DRL como solución para el problema del control de congestión en el marco del NSL arroja resultados favorables en el uso eficiente de los recursos de la red así como la eficiencia de los algoritmos anteriormente propuestos [40] [41] [42].

Algunos de los algoritmos DRL más relevantes propuestos en la literatura son: DQN [78], Red-Q Profunda Doble (DDQN, *Double Deep Q-Network*) [79], Red-Q Profunda con Reproducción de Experiencia Priorizada (Prioritized DQN, *Deep Q-Network with Prioritized Experience Replay*) [80], Red-Q Profunda de Duelo (Dueling DQN, *Dueling Deep Q-Network*) [81] y Red-Q Profunda Distribucional (Distributional DQN, *Distributional Deep Q-Network*) [82].

2.2. Trabajo Relacionado

El problema de la congestión en Internet se remonta históricamente a finales de la década de 1980. Van Jacobson en 1988 et al. [83] propuso el protocolo de transmisión TCP para manejar el problema de la congestión. Luego, TCP permitió que Internet se expandiera para soportar las crecientes demandas de tamaño y velocidad. Hoy en día, los investigadores continúan explorando nuevos enfoques para abordar el problema de la congestión basados en técnicas de ML. Algunos de estos enfoques se analizan a continuación.

2.2.1. Control de Congestión en 4G

N. Cardwell *et al.* [84] propusieron un mecanismo de control de congestión basado en la medición de los dos parámetros que caracterizan una ruta: Ancho de Banda de Cuello de Botella y Tiempo de Propagación de Ida y Vuelta (BBR, *Bottleneck Bandwidth and Round-trip propagation time*). BBR minimiza el retraso al pasar la mayor parte de su tiempo con un Producto de Retraso de Ancho de Banda (BDP, *Bandwith Delay Product*) en vuelo, al ritmo de la estimación del ancho de banda del cuello de botella. El problema general de BBR es sobrestimar el BDP. Además, BBR tiene problemas para compartir el ancho de banda de manera justa [85].

Jonathan Aina *et al.* [35] propusieron un mecanismo de control de congestión nativo basado en SDN, denominado TCP Equitativo en Centros de Información (F-DCTCP, *Fair-Data Center TCP*), centrado principalmente en un escenario de congestión (TCP *Incast*). El mecanismo proporciona mejor desempeño general en términos de rendimiento, imparcialidad y tiempo de finalización del flujo (es decir, tiempo desde que se envía el primer paquete de un flujo o el paquete SYN en TCP hasta que se recibe el último paquete [86]). Pero existen brechas en la creación de una funcionalidad de controlador autónoma y flexible para el control de la congestión y la optimización del rendimiento.

Najm, IA *et al.* [87] [88] propusieron un mecanismo llamado Protocolo de Comunicación de Capa de Transporte Mejorado (ENH-SCTP, *Enhanced Stream Control Transmission Protocol*) que reduce la duración del tiempo, para alcanzar un umbral, clasificando la ventana de congestión, rendimiento, tamaño de la cola y pérdida de paquetes como métricas de rendimiento. La implementación del mecanismo propuesto sobre la Evolución a Largo Plazo Avanzado (LTE-A, *Long Term Evolution-Advanced*) demuestra un rendimiento mejorado en términos de varios factores, como el tamaño de la ventana de congestión, el rendimiento, el tamaño de la cola y la pérdida de paquetes debido a algoritmos que mejoran el inicio lento y la congestión.

Y. Kong *et al.* [32] propusieron dos esquemas de ML (TCP Basado en Predicción de Pérdidas (LP-TCP, *Loss-Predictor based TCP*) y TCP Basado en Aprendizaje por Refuerzo (RL-TCP, *Reinforcement Learning based TCP*)) basados en el control de congestión de TCP para enlaces de cuello de botella con poco búfer en redes cableadas. Estos esquemas se basan en aprendizaje supervisado (LP-TCP) y RL (RL-TCP), cada uno de ellos comparado con variantes específicas de TCP, como TCP *NewReno* [89], TCP Basado en Aprendizaje-Q (Q-TCP, *Q-Learning based TCP*) [90], Q_a-TCP [32]. Basado en un umbral de decisión, LP-TCP ofrece una mejor compensación en el rendimiento y la demora en comparación con NewReno. Por otro lado, RL-TCP aprende de manera efectiva en entornos de red dinámicos y logra un mejor rendimiento y retraso en comparación con Q-TCP, Q_a-TCP y NewReno en varias configuraciones de red implementadas en NS-2, con variaciones en el número de nodos emisores. Los esquemas LP-TCP y RL-TCP carecen de flexibilidad en su diseño cuando se configura en entornos de red dinámicos, esto significa que su

desempeño está ligado a la sensibilidad en la configuración de los parámetros de la red.

N. Yuvaraj *et al.* [33] propusieron un esquema de asimetría de enlace de manejo de capas cruzadas (CHLA) mejorado con la prevención de congestión basada en QoS mediante el enrutamiento de la congestión. Este esquema predice el tamaño de la ventana de congestión para lograr el equilibrio de congestión en la siguiente transmisión. La predicción se realiza mediante un algoritmo basado en Máquinas de Vectores de Soporte (SVM, *Support Vector Machine*). Este algoritmo considera como atributos de entrada diversos parámetros como el factor de balance de congestión, el factor agresivo y el factor de disminución de ventana. La congestión se controla con base en el mecanismo de administración de ancho de banda. Sin embargo, debido a que se basa en el algoritmo de aprendizaje supervisado SVM, se pueden encontrar brechas ya que SVM no es adecuado para grandes conjuntos de datos. Además, su rendimiento es poco ideal si hay ruido en el conjunto de datos proporcionado.

Suyang Duan *et al.* [91] propusieron dos algoritmos de Restricción de Clase de Acceso Dinámico (D-ACB, *Dynamic Access Class Barring*) para el escenario de tráfico en ráfagas de las comunicaciones Máquina a Máquina (M2M, *Machine To Machine*) en redes 4G. El mecanismo introduce un algoritmo iterativo para actualizar adaptativamente el factor de Restricción de Clase de Acceso (ACB, *Access Class Barring*) ϕ, que produce un rendimiento casi óptimo y una reducción en el tiempo total de servicio en comparación con el esquema de Estimación de Acumulación Basada en la Deriva (DBE, *Drift-based Backlog Estimation*). Como el algoritmo es independiente del modelo de llegada de paquetes, usan los mismos parámetros en diferentes modelos de activación y obtenidos cerca a un rendimiento óptimo, lo que demuestra la robustez del algoritmo. El D-ACB propuesto para el algoritmo de asignación dinámica de preámbulos puede reducir tanto el tiempo total para servir a todos los dispositivos mMTC como el número promedio de oportunidades de acceso aleatorio requeridas por cada equipo de usuario. Este enfoque gestiona los intentos de acceso aleatorio al lado de los dispositivos MTC para reducir la congestión en una condición sobrecargada en lugar de rechazar el acceso en el Nodo B evolucionado (eNB, *evolved Node B*) o el CN. Sin embargo, los algoritmos propuestos no tienen en cuenta otros parámetros QoS importantes como la latencia, lo que hace el algoritmo poco sensible

a casos de uso donde existan dispositivos con poca tolerancia a la latencia.

Sneha Kumar Kasera *et al.* [92] propusieron y evaluaron tres mecanismos de control de congestión en la IP RAN de una red de acceso inalámbrico Acceso Múltiple por División de Código (CDMA, *Code Division Multiple Access*), control de admisión, control de diversidad y control de enrutador, para maximizar la capacidad de la red manteniendo una buena calidad de voz. Los mecanismos propuestos tienen problemas relacionados con el tráfico de datos, la congestión del enlace descendente y las RAN inalámbricas.

Chung-Ju Chang *et al.* [93] propusieron el control de la congestión utilizando técnicas difusas/neuronales para acceso múltiple integrado de división de código de secuencia directa de voz y de reserva de tramas (DS-CDMA/ FRMA) en redes celulares. Los resultados de la simulación mostraron que el controlador de congestión neuronal supera al controlador de congestión difuso.

La tabla 2.1 resume las soluciones propuestas expuestas anteriormente para 4G en el control de la congestión.

Referencia	**Descripción**	**Limitaciones de diseño**	**Escalabilidad**	**Limitaciones de simulación**
[35]	Mecanismo de control de congestión basado en SDN nativo	✓	✓	
[84]	Mecanismo de control de congestión basado en BBR	✓	✓	✓
[87] [88]	Protocolo basado en ENH-SCTP para 4G LTE-A	✓	✓	
[32]	Basado en técnicas de ML para control de congestión en TCP	✓		✓
[33]	Control de congestión basado en ML con tamaño de ventana de congestión adaptativo		✓	
[91]	Algoritmos Dynamic access class barring (D-ACB)	✓	✓	
[92]	Mecanismos de control de congestión en la IP RAN de una CDMA	✓		✓
[93]	Control de congestión usando técnicas de redes difusas/neuronales	✓	✓	

Tabla 2.1: Trabajos relacionados en 4G

2.2.2. Control de Congestión en 5G

Nathan Jay *et al.* [38] propusieron un protocolo de control de congestión basado en DRL llamado Aurora. Demostraron que la introducción del control de congestión basado en DRL permite entrenar políticas de red que capturan patrones intrincados en el tráfico de datos y las condiciones de la red. La falta de datos realistas y de apoyo para la toma de decisiones de múltiples agentes es la brecha central en este trabajo. Además, Aurora funciona mal con una latencia de 1 milisegundo porque su entorno de emulación incluye procesamiento de paquetes reales, un proceso que agrega ruido de alrededor de 1 milisegundo a la latencia, que no estuvo presente durante el entrenamiento.

Guosheng Zhu *et al.* [37] propusieron una arquitectura de aprendizaje supervisado basado en un árbol de decisión que garantiza QoS para 5G. Esta arquitectura contiene mecanismos que pueden aprender de la información y las anomalías relacionadas con métricas QoS en el pasado. Basado en dicha información, puede predecir la congestión en un entorno de alta complejidad y comportamiento de red dinámico. El detector de anomalías de QoS puede detectar la nueva anomalía con un 97 % de precisión y el resultado se puede almacenar en los datos del repositorio para volver a entrenar y mejorar el modelo. El modelo es generalmente diseñado para ser utilizado en tiempo real y en ambientes dinámicos.

Guillermo Pocovi *et al.* [94] usaron una estructura flexible de la trama de radio 5G, donde el tamaño de Intervalo de Tiempo de Transmisión (TTI, *Transmission Time Interval*) es configurable por usuario de acuerdo con sus requisitos de servicio específicos, logrando un retraso de transmisión bajo requerido para transmitir cargas útiles. Los resultados mostraron que las cargas bajas del sistema que utilizan un TTI corto (por ejemplo, 0,25 ms) son una solución para lograr comunicaciones de baja latencia. Lograr una baja latencia también reduce la congestión en la red debido al poco tiempo que se tarda en procesar los datos transmitidos. Sin embargo, los diferentes tamaños de TTI para configurar dependen del caso de uso.

Jingxuan Zhang *et al.* [95] propusieron la Estrategia de Programación de TTI Autoadaptable (SAFE-TS, *Self-Adaptive Flexible TTI Scheduling*) en los escenarios de

coexistencia eMBB y URLLC. Los resultados de la simulación demostraron reducir la latencia y la tasa de pérdida de paquetes de los servicios URLLC al tiempo que garantizan los requisitos de eMBB. En las pruebas, demostraron que el rendimiento de retraso de los servicios de URLLC mejora un 45,64 % de media, lo que alivia la congestión en la red. Las estrategias tienen problemas relacionados con la recopilación de datos en el lado de la estación base, ya que no hay otros conjuntos de datos de referencia; por lo tanto requieren de un nuevo *dataset* con el cual le den mayor confiabilidad a sus resultados.

K. Han *et al.* [40] propusieron un mecanismo que distingue las congestiones de la red y los errores inalámbricos a través de un algoritmo de DL cuando ocurre una pérdida de paquetes. En caso de pérdida por congestión de la red, el control de congestión se realiza de la misma forma que el TCP existente. En caso de pérdida debido a errores, se propone un algoritmo que puede mejorar el rendimiento de TCP inalámbrico al retransmitir solo los paquetes perdidos sin reducir la ventana de congestión. El algoritmo propuesto mejoró el rendimiento de TCP en comparación con TCP Westwood [96] o TCP Veno [97] mejorando el rendimiento de TCP inalámbrico al discriminar la congestión y el error inalámbrico. Sin embargo, las simulaciones se ejecutaban en un entorno configurado en la herramienta Network Simulator (NS-3) con parámetros ajustados, y no contemplan entornos dinámicos de red. El enfoque propuesto se centra sólo en la pérdida de paquetes y no tiene en cuenta otras métricas de QoS.

Y. Liang *et al.* [41] diseñaron una topología de simulación para analizar el rendimiento del protocolo de Notificación de Gestión Cuantizada (QCN, *Quantized Congestion Notification*) con el fin de controlar la congestión en la estación base distribuida de 5G. Los autores implementaron la topología de red con datos de 5G, cada uno con diferentes prioridades y ráfagas de alta frecuencia. Las evaluaciones se llevaron a cabo en un entorno configurado con NS-3, y demostraron que QCN puede reducir la ocupación del *buffer* en el *switch* y mantener el *buffer* cerca del umbral deseado, para mejorar la robustez de la red, y disminuir la probabilidad de la pérdida de datos. La latencia del encolado de paquetes también es significativamente reducida. Para datos con baja prioridad, QCN mejora el desempeño general de QoS en la red.

B. Han *et al.* [42] propusieron un marco de control de congestión transversal (CSCC)

que puede prever el impacto de tal decisión en el rendimiento general del sistema. Mediante el uso de técnicas de RL como Q-Learning y Algoritmos Genéticos el mecanismo permite tomar las decisiones óptimas de manera conjunta, maximizando la utilización de los recursos y garantizando que los recursos disponibles se asignan de acuerdo con las prioridades de los sectores. Este enfoque intenta maximizar los criterios de rendimiento, como la utilidad, la elasticidad, la resiliencia y la seguridad en diferentes *slices* de varios tipos. El marco de control de congestión identifica los *slices* con requisitos más flexibles para los cuales se puede reducir la cantidad de recursos asignados y actualiza su asignación de recursos.

Meysam Nasimi *et al.* [98] propusieron un mecanismo de control de congestión que opera dentro del marco de la Computación de Borde de Acceso Múltiple (MEC, *Multi-Access Edge Computing*). Para hacerlo, los autores introdujeron una función dedicada conocida como Motor de Control de Congestión (CCE, *Congestion Control Engine*), que puede capturar la condición de la RAN a través de la función de Servicio de Información de Red de Radio (RNIS, *Radio Network Information Service*) y usar este conocimiento para tomar decisiones en tiempo real. Este conocimiento se utiliza en el algoritmo propuesto para tomar la decisión de almacenar selectivamente el tráfico. Por lo tanto, el mecanismo de control de congestión puede aliviar la congestión de la red mientras hace un mejor uso de los recursos de red disponibles. Las evaluaciones están sujetas solamente a dos métricas de desempeño. La primera métrica es la probabilidad de entrega de contenido que representa cuánto tráfico se puede almacenar en el servidor de borde. La segunda métrica es el retraso en la entrega del contenido, lo que indica qué tan rápido puede ser entregado. Sin embargo, hay limitaciones de simulación debido al número máximo de servidores de borde que pueden considerar y los tiempos de plazo para cada prueba.

I.A.Najm *et al.* [99] propusieron un modelo ML basado en un algoritmo de árbol de decisiones para predecir la mejora óptima del control de congestión en los sensores inalámbricos de las redes 5G IoT. El modelo desarrollado se implementó sobre un conjunto de datos de entrenamiento para determinar la configuración paramétrica óptima en un entorno 5G, proporcionando resultados con más del 92 % de precisión y recuperación. Este modelo solo se implementó a nivel de software, es decir, en un entorno simulado.

Yi Han *et al.* [100] propusieron el algoritmo de Control de Congestión Mejorado Basado en BBR (eBCC, *enhanced BBR Congestion Control*), que reduce la tasa de pérdida de paquetes y la posibilidad de retransmisión de paquetes, al mismo tiempo que intenta mejorar la equidad del algoritmo en la transmisión de flujos múltiples. Además, cuando el enlace sufre de un mayor retraso, lograron demostrar que eBCC fue capaz de aumentar el rendimiento en la transmisión mediante la reducción de la tasa de envío y la pérdida de paquetes. Sin embargo, en la etapa de pruebas no comparan el rendimiento de la red entre diferentes flujos de algoritmos de control de congestión y diferentes flujos de RTT. Los experimentos se realizaron en una red inalámbrica simulada con NS3.

Xiang Xiao *et al.* [101] introdujeron un algoritmo de control de congestión basado en rutas cooperativas. Los autores abordan el problema del algoritmo de control de congestión basado en *MultiPath* TCP (donde un cliente puede conectarse al mismo host de destino con múltiples conexiones a través de diferentes adaptadores de red) y analiza los resultados de la simulación basados sobre la eficiencia energética del algoritmo de control de congestión. El algoritmo optimiza el mecanismo de ajuste de la ventana de congestión maximizando la energía-eficiencia de la transmisión de datos y combinando estimación de ancho de banda en tiempo real. En comparación con el clásico algoritmo de congestión de control *MultiPath* TCP, Algoritmo de Control de Congestión con Eficiencia de Energía (EECCA, *Energy-Efficiency Congestion Control Algorithm*) mantiene una alta eficiencia energética mientras que asegura ganancias de rendimiento. Los resultados experimentales arrojaron que, en comparación con *MultiPath* TCP, el rendimiento del algoritmo propuesto aumenta en un 52,6 % y tiene un mejor rendimiento. El algoritmo fue investigado en un entorno limitado de redes de sensores inalámbricos. En el caso de un entorno de red más complejo, debe buscarse un modelo para optimizar la potencia de transmisión de la red.

Geon-Hwan Kim *et al.* [102] presentaron un algoritmo que es una modificación del Protocolo de Control de Transferencia Escalable (S-TCP, *Scalable Transfer Control Protocol*) [103] que está diseñado para redes de alta velocidad. El algoritmo mejora la equidad inter-protocolo e intra-protocolo sin degradar el rendimiento de TCP en redes *mmWave*. Para lograr el objetivo de diseño 5G de alta tasa de entrega y ultra

baja latencia al mismo tiempo, el algoritmo de control de congestión TCP fue rediseñado para aumentar la utilización del enlace, y se adoptó el esquema de Gestión Activa de Colas (AQM, *Active Queue Management*) para reducir el retraso de extremo a extremo. Los autores modificando el mecanismo de aumento/disminución de la ventana de congestión de S-TCP, mejoraron la estabilidad del flujo TCP y al mismo tiempo proporcionan un alto rendimiento. Además, el esquema de gestión de colas se aplicó a las estaciones base *mmWave* para evitar el problema de *bufferfloat* (exceso de almacenamiento en búfer de paquetes), lo que reduce los retrasos de un extremo a otro.

Yaquin Song *et al.* [104] propusieron un mecanismo de control de congestión de 2 niveles. Los autores cambian la ubicación de la réplica en función de una tabla de estado de nodos para evitar rutas de congestión cuando ocurre una gran congestión. También utilizan un algoritmo de control de congestión para ajustar la tasa de envío de solicitudes, a fin de evitar la congestión del enlace en condiciones de congestión ligera. Demuestran también que reducen eficazmente el retraso de transmisión cuando se produce una gran congestión. En la evaluación del algoritmo propuesto no incluyen nodos intermedios que participen en la retroalimentación de la señal de congestión.

La tabla 2.2 resume los artículos recientes que proponen soluciones basadas en algoritmos DRL, que se considera la nueva tendencia para aplicaciones de redes como el control de congestión. En particular éste trabajo grado propone un mecanismo de control de congestión que sea escalable, es decir, fácilmente adaptable al aprovechamiento de los recursos de los nodos y del estado de los diferentes *slices*, que sea flexible, es decir, que se pueda implementar en casos de uso distintos al propuesto de URLLC, y que no posea limitaciones de simulación al ser construido en un entorno emulado.

Referencia	Descripción	Limitaciones de diseño	Escalabilidad	Flexibilidad	Limitaciones de simulación
[38]	Protocolo de control de congestión: DRL Aurora.	✓	✓		✓
[37]	Arquitectura de aprendizaje supervisado para garantizar QoS		✓	✓	
[94]	Estructura de trama flexible para 5G Radio con tamaño de TTI configurable.	✓	✓	✓	
[95]	SAFE-TS en un escenario de coexistencia de eMBB y URLLC	✓	✓		
[40]	Modelo híbrido (Algoritmo de TCP y DL)		✓		✓
[41]	Notificación de congestión cuantizada		✓		
[42]	Control de congestión intraslice		✓		✓
[98]	CCE	✓	✓		
[99]	Algoritmo de ML basado en árbol de decisiones		✓	✓	
[100]	Algoritmo de control de congestión de BBR basado en la equidad basado en QoS mediante QUIC			✓	✓
[101]	Algoritmo de control de congestión consciente de los recursos de varias etapas en un entorno de computación en la frontera	✓			✓
[102]	mmS-TCP: TCP escalable para mejorar el rendimiento y la equidad en redes 5G mmWave		✓	✓	
[104]	Mecanismo de control de congestión de dos niveles (2LCCM) para redes centradas en la información		✓	✓	✓
Este trabajo de grado	Un mecanismo de control de congestión basado en DRL		✓	✓	

Tabla 2.2: Trabajos relacionados en 5G

Capítulo 3

Sistema de Control de Congestión

Este capítulo presenta los diferentes componentes para el control de congestión propuesto. La sección 3.1 presenta una descripción general del sistema propuesto, junto con los módulos que contiene el mecanismo de control de congestión. La sección 3.2 presenta la arquitectura con sus respectivos módulos. La sección 3.3 presenta el modelado del sistema basado en DRL. Finalmente, la sección 3.4 presenta el algoritmo del mecanismo propuesto.

3.1. Descripción General

El ETSI ha desarrollado un estándar en torno a NFV que permite la configuración de las Funciones de Red Virtualizadas (VNF, *Virtual Network Function*) dentro de las infraestructuras de red en 5G [105], este estándar de Gestión y Orquestación NFV (MANO, *NFV Management Orchestration*) permite implementar un sistema de control de congestión que interactúe directamente con las VNF. A partir de MANO se implementa un mecanismo de control de congestión, el cual, en primera instancia necesita realizar un monitoreo previo de la topología del 5G CN, para recopilar información de las métricas QoS del *slice*. El ciclo de trabajo del sistema comienza cuando se extrae la información sobre el estado actual de las VNF que componen la topología del 5G CN (Unidad Central de Procesamiento (CPU, *Central Processing*

Unit), latencia, tasa de pérdida de paquetes y *jitter*). Posteriormente, el agente DRL determina si debe asignar o disminuir la cantidad de recursos disponibles para cubrir la demanda del servicio de cirugía remota. A continuación, dependiendo de la acción que toma el agente DRL, traducida en aumentar o disminuir los recursos asignados a la VNF, se valida si se cuenta con los recursos solicitados. Una vez validados, los recursos se asignan en la topología del 5G CN a la VNF que los requiera. Posteriormente, se analiza el rendimiento de las métricas de desempeño como la latencia, *jitter* y tasa de pérdida de paquetes después de la asignación de recursos en el 5GCN. Finalmente, el ciclo de trabajo del sistema inicia nuevamente.

3.2. Arquitectura

En esta sección, se presentan los módulos de la arquitectura del sistema de control de congestión para un *slice* de cirugía remota. La Figura 3.1 muestra la arquitectura de referencia con el mecanismo de control de congestión formado por el Módulo de Monitoreo, el Módulo de Control de Congestión, el Módulo de Solicitud, el Módulo de Asignación, y el Repositorio. Además, cada uno de los módulos interaccionan con los componentes de MANO, cuyos componentes principales son mencionados a continuación, junto con la interacción que tienen con el mecanismo de control de congestión propuesto.

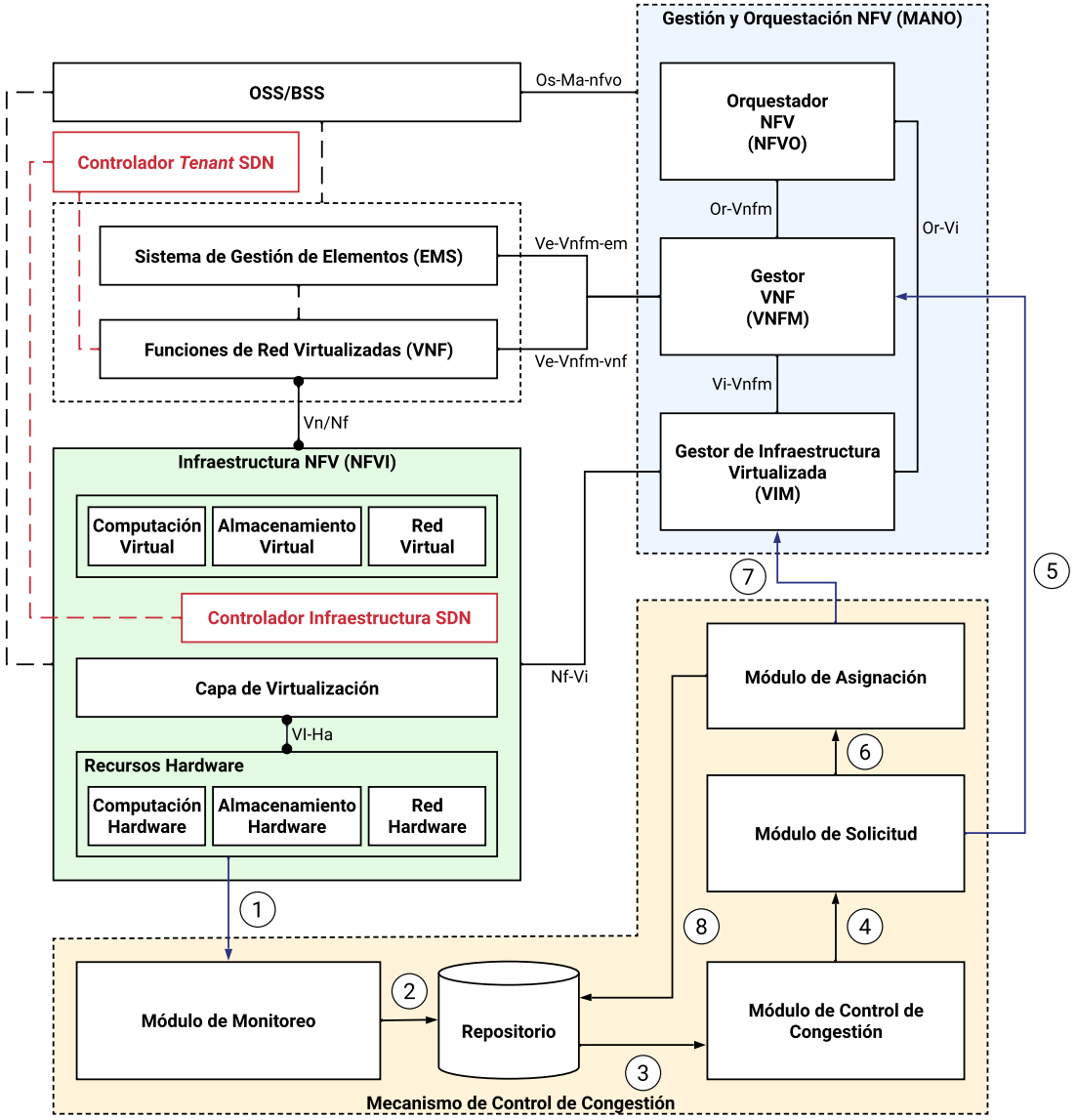

Figura 3.1: Arquitectura de referencia

3.2.1. Gestión y Orquestación de Funciones de Red Virtualizadas

MANO es el encargado de la administración del ciclo de vida de los recursos hardware, además del software que soporta la virtualización de la infraestructura y la

gestión del ciclo de vida de las VNF. MANO se centra en las tareas de gestión específicas de virtualización necesarias en el marco NFV.

Esta interactúa con el bloque OSS/BSS, lo que permite que la NFV se integre en la gestión de la red.

MANO se compone por tres bloques funcionales, estos son:

- Orquestador de Virtualización de Funciones de Red (NFVO, *Network Function Virtualization Orchestrator*)
- Gestor de Funciones de Red Virtualizadas (VNFM, *Virtual Network Function Manager*)
- Gestor de Infraestructura Virtualizada (VIM, *Virtualized Infrastructure Manage*)

Con respecto al mecanismo de control de congestión propuesto, MANO permite al mecanismo validar si se cuenta con los recursos necesarios para suplir la demanda de recursos solicitada por la VNF del 5G CN.

3.2.1.1. Orquestador de Virtualización de Funciones de Red

NFVO es responsable del despliegue de nuevos servicios de red, de la gestión del ciclo de vida de los servicios de red y la gestión de recursos globales (recursos virtuales, reservas de recursos, rendimiento de los recursos, fallos de recursos, etc). NFVO además, está encargado del proceso de validación y autorización de peticiones de NF de recursos para la Infraestructura de Virtualización de las Funciones de Red (NFVI, *Network Function Virtualization Infrastructure*), seguimiento de disponibilidad y la asignación de recursos a través del VIM. Este bloque es el encargado de gestionar la asignación en el mecanismo de control de congestión de los recursos de unidades de CPU para la VNF que demanda recursos.

3.2.1.2. Gestor de Funciones de Red Virtualizadas

VNFM administra la gestión del ciclo de vida de las VNF. Se encarga de recopilar y ofrecer información sobre el comportamiento de las VNF del 5G CN al Módulo de Solicitud, con el fin de validar con qué recursos cuentan las VNF de la topología. Además, es el responsable de la coordinación y adaptación para la configuración y la presentación de informes de eventos entre la NFVI y el Sistema de Gestión de Elementos (EMS, *Element Management System*).

3.2.1.3. Gestor de Infraestructura Virtualizada

VIM controla y administra los recursos de computo, almacenamiento y red de la NFVI para que la arquitectura MANO funcione correcta y eficazmente, debe integrarse mediante interfaces (Ver tabla 3.1) para interactuar con la capa de virtualización de la NFVI. Por otro lado, un VIM puede estar especializado en el manejo de un cierto tipo de recurso de NFVI (e.g., solo computación, solo almacenamiento o solo red), o puede ser capaz de administrar múltiples tipos de recursos. Este bloque se encarga de recibir las solicitudes del Módulo de Asignación, ya que al tener comunicación directa con la NFVI, mediante la interfaz Nf-Vi permite asignar los recursos demandados y gestionar su respectivo despliegue en la VNF del 5G CN.

3.2.2. Sistemas Operativos y de Soporte Empresarial

OSS / BSS son implementados por el proveedor de servicios de VNF [106]. El ETSI no nombra a OSS y BSS como un componente dedicado a NFV, sin embargo, OSS y BSS solicitan servicios de una instancia NFV y los consumen.

El OSS / BSS incluye la colección de sistemas y aplicaciones de gestión que los proveedores de servicios utilizan para operar su negocio, además de funciones de MANO. En el mecanismo de control de congestión propuesto, este bloque no presenta gran impacto debido a la carencia de una implementación tipo comercial que se despliegue.

Interfaz	Ubicada entre	Descripción
Vi-Ha	Capa de Virtualización y Recursos Hardware dentro de la NFVI	Descubrir y recopilar información de configuración y sus recursos. Crear un entorno de ejecución (e.g., VNF) para cargas de trabajo.
Vn-Nf	NFVI y VNF	Aquí la VNF representa el entorno de ejecución. La interfaz se utiliza para especificar interacciones entre los aceleradores VNF y NFVI. Las interfaces se pueden utilizar para descubrir, configurar y administrar estos aceleradores y para que la VNF cancele el registro para recibir eventos y datos del acelerador.
Nf-Vi	NFVI y VIM	Descubrir / recopilar recursos físicos / virtuales y su información de configuración. Administrar (crear, redimensionar, (des) suspender, reiniciar, etc.) recursos físicos / virtualizados. Cambios en la configuración de recursos físicos / virtuales. Configuración de recursos físicos / virtuales.

Tabla 3.1: Puntos de referencia

3.2.3. Funciones de Red Virtualizadas y Sistema de Gestión de Elementos

Administrada por el VNFM, es donde se sitúan las diferentes VNF implementadas en software para ejecutarse en recursos virtuales de computación, almacenamiento y red, junto con sus respectivos EMS que gestionan las VNF si es el caso. La VNF es la entidad correspondiente a todos los nodos de red y se espera que se entregue únicamente como software dependiente del hardware [105].

3.2.4. Infraestructura de Funciones de Red Virtualizadas

NFVI es la encargada de comprender los recursos hardware y software que crean el entorno en el que se despliegan las VNF. Virtualizando la computación física, el almacenamiento y las redes, agrupándolos en grupos de recursos. Esta proporciona los recursos necesarios para soportar la ejecución de las VNF [105].

La NFVI proporciona una infraestructura de múltiples usuarios aprovechando la tecnología de virtualización de las tecnologías de la información que pueden soportar múltiples casos de uso [105]. La NFVI abarca tres dominios:

- **Dominio de cómputo:** proporciona servidores y almacenamiento de alto volumen comercial.
- **Dominio del hipervisor:** gestiona los recursos del dominio de cálculo a las máquinas virtuales de los dispositivos de software, proporcionando una abstracción del hardware.
- **Dominio de la red de infraestructura:** comprende todos los conmutadores genéricos de alto volumen interconectados en una red que se puede configurar para suministrar servicios de red de infraestructura.

3.2.5. Mecanismo de Control de Congestión

El enfoque del control de congestión se implementa en el 5G CN, el cual está compuesto por diferentes VNF (ver Figura 3.2) que siguen el concepto de Separación de Planos de Control y de Usuario (CUPS, *Control and User Plane Separation*). CUPS define la separación entre el Plano de Control (CP, *Control Plane*) y el Plano de Usuario (UP, *User Plane*) en el 5G CN para una implementación flexible, evolución independiente y escalable [107]. Cada grafo incluye una CP VNF compuesta por una Función de Gestión de Acceso y Movilidad (AMF, *Access and Mobility Function*) y una Función de Gestión de Sesión (SMF, *Session Management Function*), y una UP VNF compuesta por una Función de Plano de Usuario (UPF, *User Plane Function*)

[107] la cual es la encargada de recibir y retransmitir los paquetes de datos en el 5G CN, siendo este el punto de interés para un control de congestión eficiente. Por otra parte el grafo de URLLC considera las copias de seguridad para cada AMF, SMF y UPF que deben instanciarse en diferentes nodos cercanos al usuario final, ya que URLLC debe ofrecer alta confiabilidad y baja latencia [108].

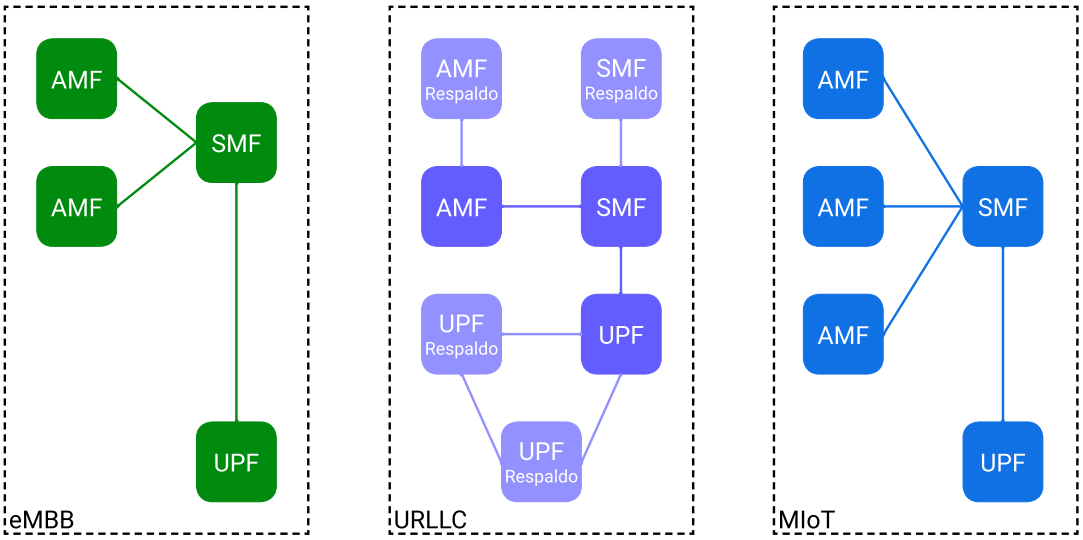

Figura 3.2: Caracterización de grafos para cada tipo de servicio

- AMF: es una de las NF del plano de control del 5G CN. El 5G AMF, es una evolución de la Entidad de Gestión de Movilidad (MME, *Mobility Management Entity*) en 4G, continuando con el Plano de Control y *User Plane Separation*, y con más simplificaciones como mover las funciones de administración de sesiones a la SMF y, proporcionando interfaces de Arquitectura basada en Servicios (SBA, *Service-Based Architecture*) comunes [109].
- SMF: es responsable del manejo de sesión con las funciones individuales soportadas por sesión. Interactúa con el plano de datos desacoplado, crea sesiones de actualización y eliminación de unidades de datos de protocolo y administra el contexto de la sesión con la UPF [110].
- UPF: representa la evolución del plano de datos de una estrategia de CUPS, introducida como una extensión de los Núcleos de Paquetes Evolucionado (EPC,

Evolved Packet Core). CUPS desacopla el control de la Puerta de Enlace de Paquetes (PGW, *Packet Gateway*) y las funciones del plano de usuario, lo que permite descentralizar el componente de reenvío de datos. Esto permite que el procesamiento de paquetes y la agregación de tráfico se realicen más cerca del borde de la red, lo que aumenta la eficiencia del ancho de banda [111].

A continuación, se describen las funciones de cada uno de los módulos que componen el mecanismo de control de congestión:

- **Módulo de Monitoreo**: lleva a cabo la primera tarea de la arquitectura del mecanismo de control de congestión. En esta tarea se analiza la topología del 5G CN y se recolecta la información de métricas importantes tales como la latencia, la tasa de perdida de paquetes y el *jitter*, además, del consumo de CPU. Las métricas se extraen desde los Recursos Hardware de NFVI ①. Finalmente, una vez obtenidas las métricas, se almacena la información en el **Repositorio** ②.
- **Módulo de Control de Congestión**: recibe la información del estado actual de la infraestructura del 5G CN ③ y permite la entrada de un conjunto de métricas previamente almacenadas en el **Repositorio**, que representan los elementos de entrada de la arquitectura propuesta para el mecanismo de control de congestión. El Módulo de Control de Congestión incluye el Agente DRL, este se encarga de la toma de decisiones mas óptima con base en la experiencia anterior y al estado actual según los índices de usabilidad y disposición de los recursos de unidades de CPU que satisfagan los requisitos de QoS, principalmente de latencia (ver sección 3.3), esta decisión se traduce en cambios en los recursos de unidades de CPU asignadas a la UPF, la cual se encarga de procesar el trafico de datos en el 5G CN. Por otra parte, el Módulo de Control de Congestión se encarga de asignar los recurso de unidades de CPU eficientemente, evitando la sobreasignación y el desperdicio de este recurso con el fin de ser aprovechado por otro *slice*.
- **Módulo de Solicitud de Recursos**: actúa en consonancia con la salida del Agente DRL que representa una acción. Produce por lo tanto, una solicitud de

recursos en caso de que sean demandados para el cumplimiento del requisito de latencia del 5G CN. La solicitud de recursos se realiza al VNFM ⑤, el cual debe validar la existencia, para su posterior solicitud de asignación de recursos al **Módulo de Asignación de Recursos** requeridos por el *slice* ⑥. Mediante la interfaz Vi-Vnfm se intercambian elementos de información entre el VIM y el VNFM. Dichos elementos corresponden según el ETSI a recursos de cómputo, de red y de almacenamiento virtualizados.

- **Módulo de Asignación de Recursos**: recibe la solicitud de recursos de ⑥ y localiza los recursos mediante el VIM ⑦, que asigna los recursos demandados en la topología del 5G CN.

3.3. Agente de Control de Congestión

Un agente DRL se encarga de encontrar la política de decisión óptima, que maximiza la recompensa y como consecuencia logra controlar la congestión haciendo un uso eficiente de los recursos asignados a cada VNF. Luego, el agente DRL es capaz de maximizar el rendimiento de la red en situaciones en las que se ven involucradas un numero determinado de cirugías remotas simultaneas, debido a que el criterio de decisión se actualiza en función de los estados pasados y la expectativa de los estados futuros, en relación a los recursos asignados a la VNF.

A diferencia de los algoritmos propuestos en la literatura, el agente DRL propuesto para desarrollar este mecanismo de control de congestión se basa en la asignación eficiente de recursos computacionales en los nodos para maximizar la recompensa, obtener un aprendizaje acelerado y reducir la latencia.

Inicialmente, la red es representada por $G = (N, L)$, donde N representa todo el conjunto de nodos y L todo el conjunto de enlaces. Los nodos virtuales N_n están asociados con recursos de CPU, modelados como $Res_{CPU}(N_n)$.

Cada enlace en la red está dado por $L_{i,j}$ y corresponde al enlace dado de N_i y N_j. La latencia para el enlace $L_{i,j}$ se indica como $La(L_{i,j})$, la perdida de paquetes como $Pl(L_{i,j})$ y el *jitter* como $Ji(L_{i,j})$

Dado que los *slices* representan diferentes casos de uso aparte de la cirugía remota, como la comunicación vehicular y la IoT, estos requieren diferente cantidad de recursos. SDN y NFV, hace que cada nodo N_n despliegue los VNF correspondientes, y que cada enlace $L_{i,j}$ se divida en múltiples enlaces virtuales que se pueden compartir entre *slices*. Como resultado, el control de congestión implica la administración de los recursos asignados a los nodos y como consecuencia del estado de los enlaces. Un controlador centralizado se encarga de cumplir con los requerimientos principalmente de latencia del *slice* de cirugía remota. Al monitorear la red, el controlador puede asignar dinámicamente los recursos asociados a cada VNF de cada nodo N_n.

El controlador maneja una cola de solicitudes, cada una de las cuales solicita una cantidad de recursos diferentes durante un período de tiempo, y gestiona los recursos según los requisitos del *slice* de cirugía remota, tomando en cuenta que cada intervalo de tiempo t llegan nuevas solicitudes del *slice*, y el mecanismo ajusta dinámicamente los recursos asignados.

Los recursos de unidades de CPU obtenidas del *slice* de cirugía remota se representan como un vector $ob_n(m)$, dado por:

$$ob_n(m) = [ar_{m,n}, ur_{m,n}]$$

Donde $ar_{m,n}$ denota los recursos asignados del tipo de recurso m actualmente asignados a la n-ésima VNF en el nodo N_n, y $ur_{m,n}$ denota los recursos usados.

3.3.1. Espacio de Estados

En el mecanismo, se restringen los porcentajes para que oscilen entre el 0 % y el 100 % del total disponible del tipo de recurso de CPU. Si el nivel de precisión para el control es del 1 %, se tienen 100 estados diferentes que van de $[1\,\%, 2\,\%, 3\,\%, ..., 99\,\%, 100\,\%]$. En este caso, para un nodo N_n en el 5G CN, la matriz de espacio de estados será de $2 * n$ filas que corresponden a los recursos asignados y usados dado el recurso de CPU por cada VNF en el nodo y 100 columnas correspondientes a los estados. Por lo tanto, la matriz de espacio de estados del *slice* s será de $[2 * n][100]$ estados.

3.3.2. Espacio de Acciones

Las acciones de salida generadas por el agente y ejecutadas por el controlador indican el ajuste de los recursos asignados a cada VNF del correspondiente nodo, dado por:

$$A = \{a_c | a_c = x * c\}, c = \{0, 1, ..., z\}, x \in \{-1, 1\} \tag{3.1}$$

Donde la acción es representado por a_c donde c es el porcentaje de recurso del total disponible y x indica si aumenta o disminuye según el valor (Ver Algoritmo 1).

Algoritmo 1: Generador de espacio de acciones

Entrada : Límite de ajuste z
Salida : Conjunto de acciones A

```
for a_c to A do
    Se establece la acción indicando el valor c
    Se establecer si es incremental o decremental con el valor x
    Se genera el valor de ajuste del recurso dado para a_c
    a_c = x * c
end
```

3.3.3. Recompensa

Al abordar el problema de asignación de recursos, el objetivo general es minimizar la latencia del *slice* de cirugía remota, garantizar el rendimiento y maximizar la utilización de recursos de los nodos físicos. Específicamente, la función de recompensa está diseñada para relacionarse con el rendimiento del *slice* en términos de la utilización de los recursos de la red. El rendimiento del *slice* depende de cada VNF que se implementa en diferentes nodos. Por lo que la disminución del rendimiento de cualquier VNF afectaría el rendimiento general del *slice*. Es por eso que la recompensa P del *slice* de cirugía remota se define para combinar dos partes: garantizar la latencia en el *slice* de cirugía remota y evitar el desperdicio de recursos, la cual está dada por:

$$P(s) = \begin{cases} \theta * \frac{ur_m+\nu}{ar_m}, & si\ ar_m > ur_m + \nu \\ \eta * e^{vio(s)}, & en\ otro\ caso \end{cases} \tag{3.2}$$

En el primer caso se indica que el recurso asignado es suficiente para satisfacer el minino recurso viable en la VNF, siendo la recompensa inversamente proporcional al recurso asignado (ar_m) con respecto al usado (ur_m), logrando así una recompensa óptica en cuanto al desperdicio de recursos, donde ν denota el umbral de prevención. En otro caso con $\eta * e^{vio(s)}$ como penalización por la violación del requerimiento, donde, $vio(s)$ denota el número de violaciones del requerimiento en el *slice* de cirugía remota en cada VNF. El exponente se utiliza para modelar el impacto de múltiples VNF con recursos insuficientes en el *slice*. Por otra parte, las constantes de penalización θ y η simplemente son los pesos de los objetivos óptimos, y se pueden ajustar según el servicio.

3.3.4. Redes Neuronales

El agente DRL propuesto utiliza DQN, lo cual le permite reducir la carga computacional y mejorar su capacidad predictiva. En la arquitectura del agente propuesta (ver Figura 3.3) se utilizan dos ANNs; la ANN Objetivo y la ANN Evaluativo. La ANN Evaluativo se utiliza para calcular los valores Q de los estados actuales, entrenándose en cada iteración buscando siempre reducir la Pérdida (es decir, lo alejado que se está de la política óptima). Por otra parte, la ANN Objetivo estima los Valores Q Objetivo los cuales calculan en sí la Pérdida, facilitando así el entrenamiento de la ANN Evaluativo. Además, los pesos de la ANN Evaluativo se transfieren a la ANN Objetivo frecuentemente, esto con el fin de que las estimaciones de la ANN Objetivo se actualicen y el agente DRL mejore. Además, el agente DRL almacena las experiencias previas en la Memoria de Repetición con cada iteración de entrenamiento, esto con el fin de que el proceso de aprendizaje disminuya los tiempos de convergencia, la varianza de aprendizaje y la correlación entre muestras consecutivas [112].

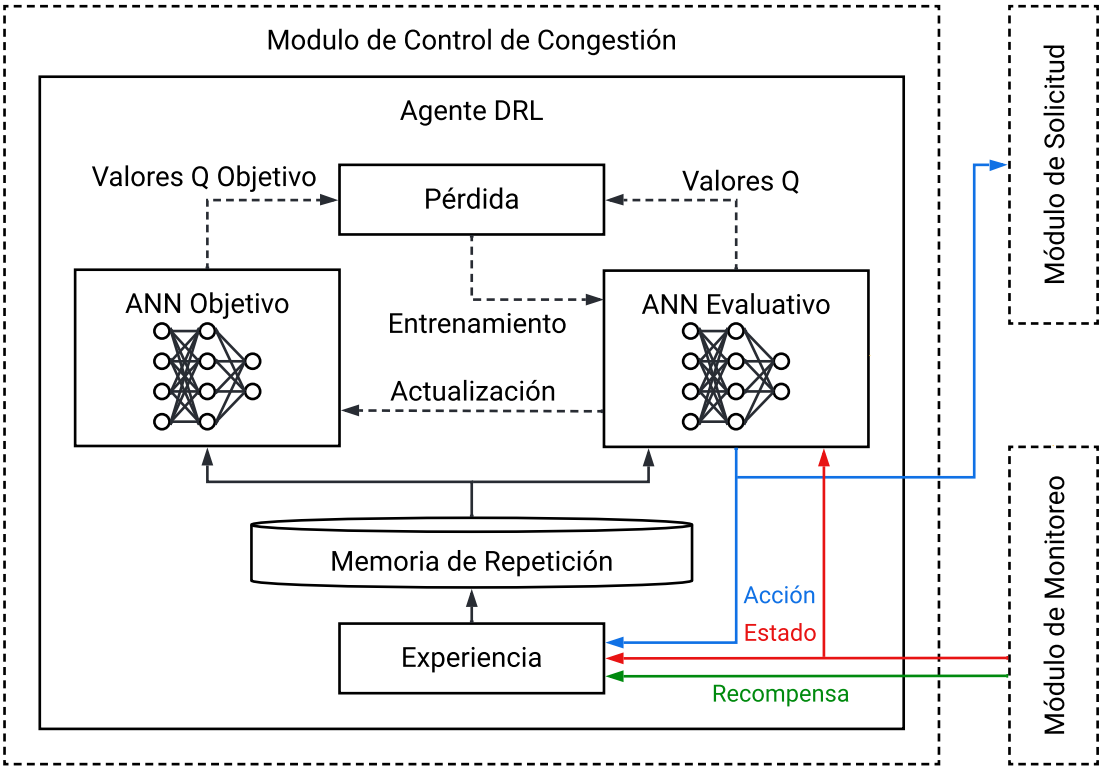

Figura 3.3: Arquitectura agente DRL

3.3.5. Exploración y Explotación

Puesto que el Deep Q-Learning cuando encuentra una buena política, sigue ejecutando dicha política mediante la explotación, es de suma importancia que el agente en ocasiones explore nuevas opciones y en otras explote las ya encontradas, es por ello que se implementa el algoritmo $\epsilon - greedy$, el cual garantiza que cada cierto tiempo la tasa de aprendizaje vaya decreciendo. Por ejemplo, si el valor de ϵ es 0.1, el algoritmo optará por explotar el 90 por ciento del tiempo y explorar solo el 10 por ciento. El agente sigue la Ecuación 3.3 para seleccionar la siguiente acción en un estado específico.

$$Acción\ en\ un\ tiempo\ (t) = \begin{cases} maxQ_t(A) & con\ probabilidad\ \ 1-\epsilon \\ cualquier\ acción\ (A) & con\ probabilidad\ \ \epsilon \end{cases} \tag{3.3}$$

3.4. Algoritmo de Control de Congestión

El Algoritmo 2 describe el mecanismo de control de congestión basado en DRL propuesto, enfocado en la optimización de los recursos computacionales. El algoritmo recibe como parámetros de entrada; el espacio de estados S y el espacio de acciones A desarrollados en un entorno emulado, además del número de episodios de aprendizaje n, el número de pasos por episodio m y el parámetro de exploración ϵ. Por otra parte, la salida es la política π la cual se enfoca en la reasignación de recursos con el fin de controlar la congestión y por ende mitigar la latencia.

El Algoritmo 2 empieza obteniendo el modelo (línea 2), el cual crea la ANN con sus capas de entrada y salida (línea 3), además de agregar las capas ocultas de dicha ANN (línea 4). Luego se obtiene el agente (línea 5), es decir, crea la política (línea 6 y 7), la estructura de datos para la memoria de repetición (línea 8) y la DQN (línea 9), descritos en la sección 3.3. A partir de aquí el algoritmo obtiene el estado inicial (línea 10) entregado por el módulo de monitoreo, el cual extrae las métricas de la red en tiempo real, tales como la latencia, los paquetes perdidos y el *jitter*, además de la asignación y el uso de unidades de CPU de la VNF, siendo la asignación y uso de las unidades de CPU en la VNF cruciales para la toma de decisión del algoritmo (ver sección 3.2.5). Una vez obtenido el estado inicial se actualiza según la política π (línea 11). Luego, el algoritmo entra a un ciclo en el cual se indica el conjunto de m pasos para la toma de decisión final (línea 12). Se selecciona una acción del espacio de acciones A (línea 13) con su respectiva recompensa $P(s)$ dado el estado (línea 14) y obteniendo como resultado la minimización de la función de pérdida (línea 16), recolectando y almacenando la experiencia en la memoria de repetición previamente (línea 15). Finalmente, la acción seleccionada lleva al agente a un nuevo estado (línea 17), repitiendo así el bucle según los n de episodios indicados. Finalmente se retorna

la política π resultante (línea 20).

Algoritmo 2: Algoritmo de Control de Congestión basado en DRL

Entrada : Número de episodios de aprendizaje: n
Número de pasos por episodio: m
Parámetro de exploración: ϵ_i

Salida : Política π

1 **for** *episodios* **to** n **do**
2 **Obtiene el modelo:**
3 Crea la ANN
4 Agrega capas ocultas
5 **Obtiene el agente:**
6 $\epsilon \leftarrow \epsilon_i$
7 política $\pi \leftarrow \epsilon$-greedy(Q) (ver Ecuación 3.3)
8 Crea memoria de repetición
9 Crea la DQN
10 Obtiene el estado inicial s_0
11 estado actual $s_t \leftarrow$ estado inicial s_0, usando π
12 **for** *pasos* **to** m **do**
13 Selecciona una acción A (ver Ecuación 3.1)
14 Obtiene una recompensa $P(s)$ (ver Ecuación 3.2)
15 Recolecta y Almacena la experiencia en la memoria de repetición
16 Se minimiza la función de pérdida
17 estado actual $s_t \leftarrow$ estado futuro s_{t+1}, usando π
18 **end**
19 **end**
20 **return** π

Capítulo 4

Evaluación y Análisis de Resultados

4.1. Evaluación

En esta sección se presenta la evaluación del mecanismo de Control de Congestión, con el fin de observar su eficiencia en términos de satisfacción del requisito de QoS de latencia, perdida de paquetes y *jitter*. La sección 4.1.1 presenta el entorno de pruebas y los requisitos de latencia, perdida de paquetes y *jitter* del *slice*. La sección 4.1.2 define las métricas de evaluación. Finalmente, la sección 4.1.3 presenta los resultados de las variaciones de los parámetros mencionados en la sección 4.1.2.

4.1.1. Entorno de Evaluación

El entorno de evaluación establece requisitos específicos de QoS en términos de latencia, perdida de paquetes y *jitter* en el 5G CN para un *slice*. El servicio de cirugía remota requiere de una demanda de latencia estricta de 0.1 milisegundos, de perdida de paquetes por debajo del 1 % y de *jitter* menor a 30 milisegundos [8] [13] [14].

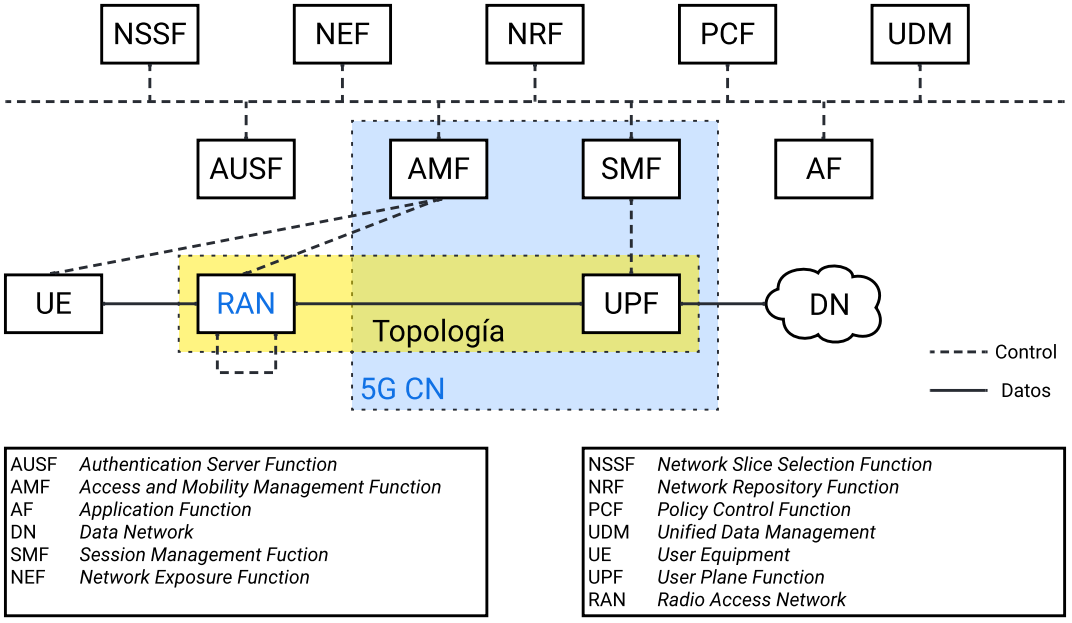

Figura 4.1: Topología de evaluación

Para la evaluación se plantea una topología de 2 nodos que representan la RAN y el 5G CN (ver Figura 4.1), donde, en el caso del 5G CN se emula la UPF y es aquí donde actúa el módulo de control de congestión, dado que al tratarse de una VNF es posible ajustar los recursos de CPU según los requerimientos de QoS solicitados por el *slice* (como se presenta en el capítulo 3), es importante aclarar que se emula la UPF debido a que está conectada mediante un enlace de datos a la RAN y por ende, es ahí donde se presenta el trafico de red. Estos nodos se crean a partir de la herramienta Containernet, que permite usar contenedores Docker como hosts en topologías de red emuladas. Los procesos de construcción, simulación y evaluación de los distintos componentes del mecanismo se desarrollan a través del lenguaje de programación Python (ver Figura 4.2). Estos procesos se ejecutan en una máquina con sistema operativo Ubuntu 18.04 con 16 GB de memoria RAM, y con una CPU AMD Ryzen 5 2600. La creación, implementación, monitorización y despliegue del *slice* se emula utilizando Containernet. La generación de tráfico de cirugía remota se realiza usando Generador de Tráfico de Internet Distribuido (D-ITG, *Distributed Internet Traffic Generator*).

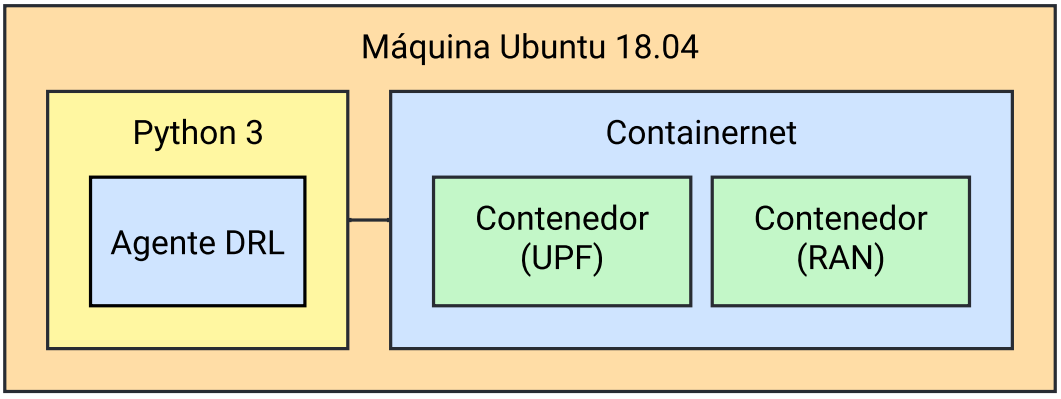

Figura 4.2: Arquitectura de ambiente de experimentación

El mecanismo de control de congestión se basa en 2 archivos Python: *monitoring.py* y *algoritmo.py*. El archivo *monitoring.py* se encarga de crear la topología de red con los recursos de CPU iniciales, por otra parte, genera el trafico de red equivalente al numero de cirugías remotas simultaneas establecidas con el uso de D-ITG, además, se encarga de obtener las diferentes métricas de latencia, perdida de paquetes y *jitter*, así como los recursos de CPU asignados y usados. Esta información se envía al archivo *algoritmo.py* (estado actual), el cual se encarga de realizar el modelado del espacio de estados, espacio de acciones y su respectiva función de recompensa, además de contener la lógica del agente DRL y sus componentes (como se enfatiza en la sección 3.4). Finalmente, *monitoring.py* contiene la función de actualizar los requisitos de CPU mediante el uso de Docker.

4.1.2. Métricas

En este trabajo de grado se establecen tres métricas para determinar el desempeño del mecanismo de control de congestión propuesto:

- Latencia.
- Pérdida de paquetes

- *Jitter*

Estas métricas se obtienen en diferentes escenarios al variar el número de cirugías remotas simultaneas y la cantidad de unidades de CPU iniciales en la UPF (ver Tabla 4.1), lo que permite visualizar cómo responde el algoritmo frente a situaciones donde se presenta congestión debido a la alta cantidad de tráfico generado, además de evaluar cómo el mecanismo de control de congestión maneja la asignación de recursos eficientemente.

Unidades de CPU (UPF)	**Número de Cirugías Remotas**
1	1
3	2
5	3
7	4

Tabla 4.1: Variación de parámetros de unidades de CPU y número de cirugías remotas

4.1.3. Experimentación

El mecanismo de control de congestión propuesto, comprende unos parámetros específicos de unidades de CPU y número de cirugías remotas simultaneas. En el entorno de pruebas, se tienen disponibles 8 unidades de CPU. Se proponen 16 escenarios para observar el comportamiento del mecanismo y el impacto que tiene en el requisito de las métricas indicadas en la sección 4.1.2. Se mantiene fijo el número de unidades de CPU asignados al nodo que corresponde a la RAN debido a la capacidad limitada que tiene la herramienta D-ITG al momento de generar el tráfico, además se mantiene la memoria asignada en 512 MB, debido a que esta no supera las 20 MB en ningún momento (ver Figura 4.3). Se varían entonces el número de unidades de CPU iniciales asignadas en el nodo que corresponde a la UPF con el fin de visualizar situaciones de asignación eficiente de recursos. Además, para cada variación se aumenta el número de cirugías remotas desde 1 hasta 4, debido a la limitada capacidad de la herramienta D-ITG y de la máquina en donde se lleva a cabo el experimento.

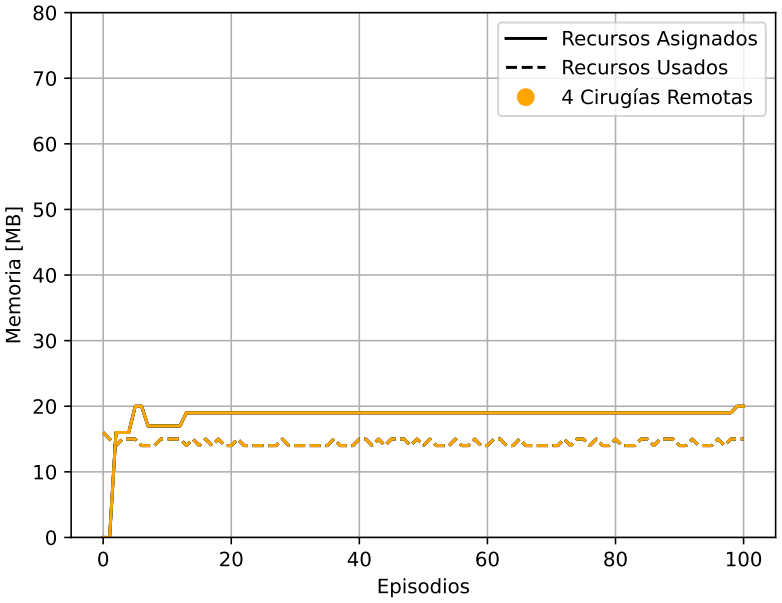

Figura 4.3: Gráfica de recursos relacionados con la memoria

Por otra parte, el número de cirugías remotas emula solicitudes de tráfico caracterizadas en la tabla 4.2.

Número	Tráfico Generado	Tipo de tráfico	Tasa de transmisión
1	Video 3D	UDP	1.6 Gbps
2	EMG	TCP	1.536 Mbps
3	Flujo de Audio	UDP	200 Kbps
4	Retroalimentación háptica	TCP	400Kbps
5	Presión de sangre, tasa de latidos, tasa de respiración	TCP	12 Kbps

Tabla 4.2: Caracterización de tráfico de una cirugía remota

Inicialmente se varían los parámetros del número de cirugías remotas y unidades de CPU iniciales, ya que ambos parámetros controlan el impacto en la latencia, el *jitter* y la tasa de pérdida de paquetes. Las Figuras 4.4, 4.5, 4.6 y 4.7 muestran que la tasa de asignación de recursos responde proporcionalmente a la demanda de

recursos usados. En la Figura 4.4 se observa que los recursos asignados inicialmente no son suficientes para soportar el trafico de mas de 1 cirugía remota sin perjudicar los requisitos de QoS, por lo que se evidencia el aumento de los recursos asignados puesto que los recursos usados ocupan toda la capacidad de los recursos asignados en la UPF. De manera similar, en la Figura 4.5 se observa que los recursos asignados inicialmente no son suficientes para soportar el trafico de mas de 1 cirugía remota, por lo que del mismo modo se evidencia el aumento de los recursos asignados. Por otra parte, en la Figura 4.6 se muestra que los recursos asignados inicialmente son suficientes para soportar el trafico de 1 y 2 cirugías remotas, sin embargo en estos casos se presenta una sobreasignación de recursos por lo que se reducen los recursos asignados en pro del uso eficiente de los mismos. De igual forma, en la 4.7 se observa como los recursos asignados inicialmente son suficientes para soportar el trafico de hasta 4 cirugías remotas por lo que del mismo modo el mecanismo de control de congestión disminuye los recursos asignados.

Figura 4.4: Recursos asignados y usados con 1 unidad de CPU inicial

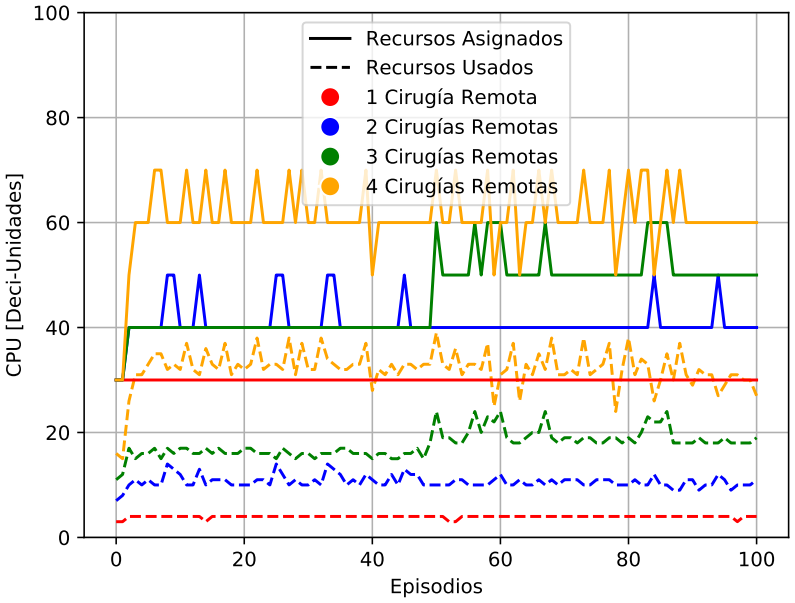

Figura 4.5: Recursos asignados y usados con 3 unidades de CPU iniciales

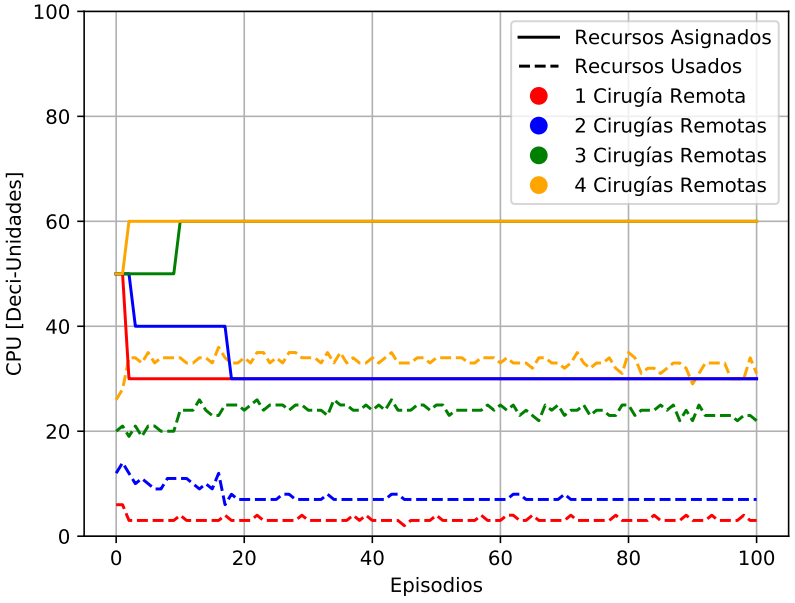

Figura 4.6: Recursos asignados y usados con 5 unidades de CPU iniciales

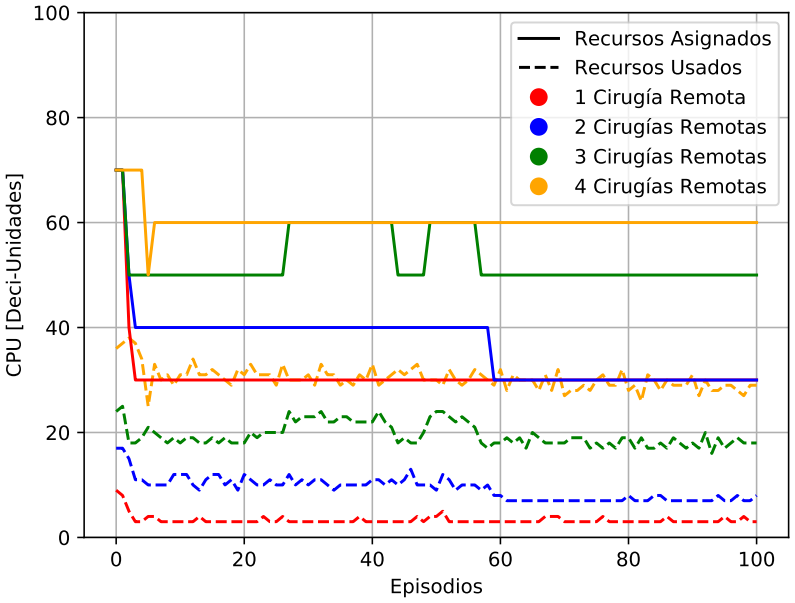

Figura 4.7: Recursos asignados y usados con 7 unidades de CPU iniciales

4.2. Resultados y Análisis

En esta sección el mecanismo de control de congestión basado en DRL es evaluado en términos de las métricas propuestas en la sección 4.1.2.

4.2.1. Latencia

Las Figuras 4.8 4.9 4.10 y 4.11 muestran los resultados de latencia para los 4 escenarios de unidades de CPU iniciales para cumplir con el requisito establecido de 0.1 milisegundos en el 5G CN, los resultados se muestran en la Tabla 4.3.

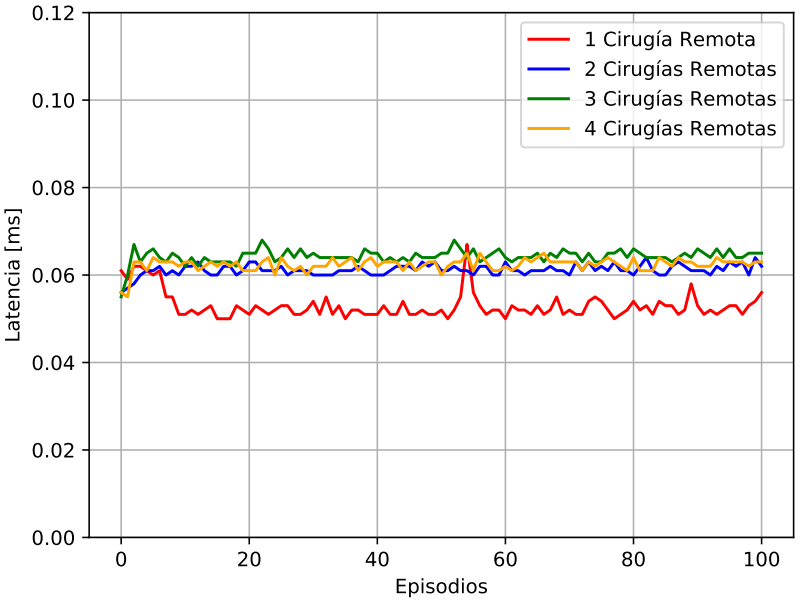

Figura 4.8: Latencia con 1 unidad de CPU inicial

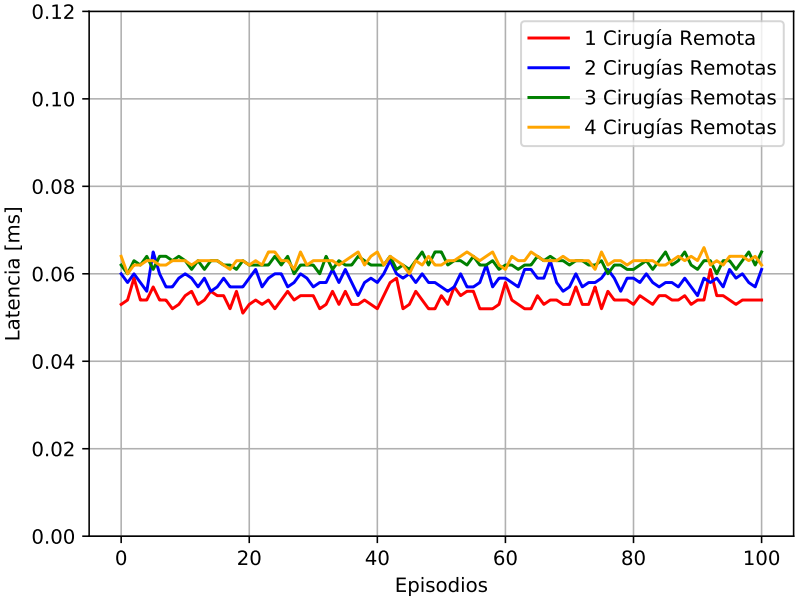

Figura 4.9: Latencia con 3 unidades de CPU iniciales

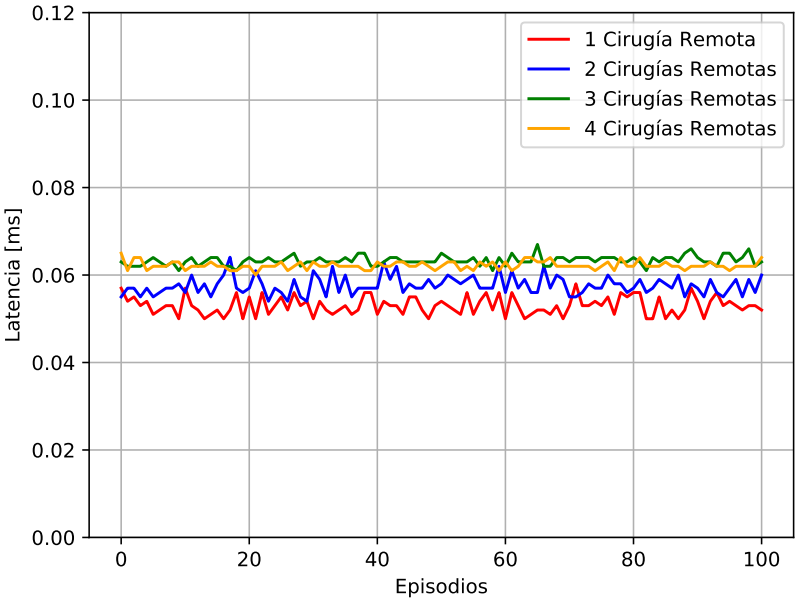

Figura 4.10: Latencia con 5 unidades de CPU iniciales

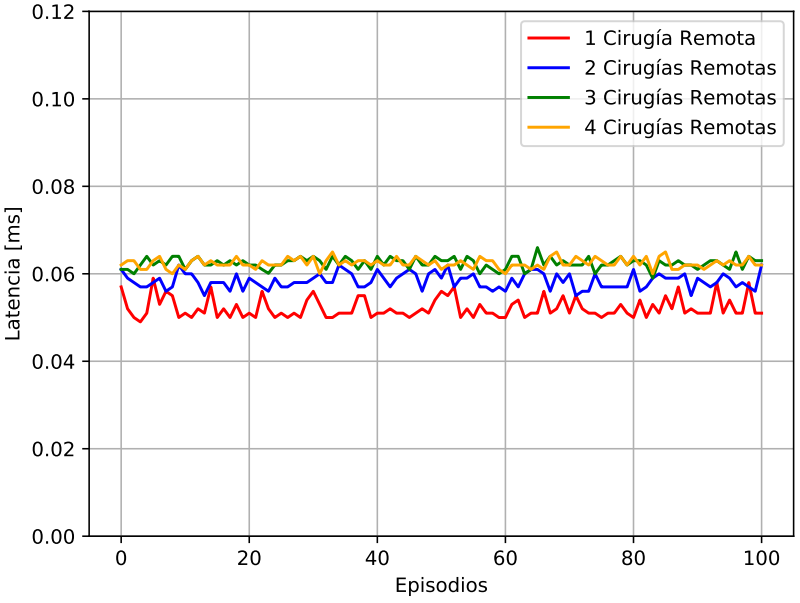

Figura 4.11: Latencia con 7 unidades de CPU iniciales

En el caso de 1 unidad de CPU inicial (ver Figura 4.8) la latencia promedio es de 0.06018 milisegundos, un 40 % distante del umbral establecido. De manera similar, en el caso de 3 unidades de CPU iniciales (ver Figura 4.9) la latencia promedio es de 0.05955 milisegundos, un 41 % distante del umbral establecido. Por otra parte, en el caso de 5 unidades de CPU iniciales (ver Figura 4.10) la latencia promedio es de 0.05895 milisegundos, un 41 % distante del umbral establecido. Finalmente para 7 unidades de CPU iniciales (ver Figura 4.11) la latencia promedio es de 0.0588 milisegundos, un 41 % distante del umbral establecido.

Unidades de CPU (UPF)	**Latencia promedio [ms]**
1	0.06018
3	0.05955
5	0.05895
7	0.0588

Tabla 4.3: Resultados de latencia

Estos resultados evidencian el hecho de que al aumentar el número de cirugías remotas, la latencia tiende también al aumento debido a la congestión que representa la generación de más tráfico. Por lo tanto, el mecanismo de control de congestión frente a esta situación responde asignando recursos de CPU necesarios para estabilizar la latencia, debajo del umbral establecido de 0.1 milisegundos. Debido a que en la caracterización de tráfico expuesta en la Tabla 4.2 existe envío de paquetes TCP, es fundamental que los datos lleguen correctamente al destinatario, sin errores y en orden. Un factor importante que influye en la latencia de la red son el tamaño de los paquetes transmitidos. Hay que tener en cuenta que existen flujos ratones y elefante como se muestran en la Tabla 4.2. Este tipo de flujo elefante que se ejecuta durante un tiempo relativo de la prueba consume una gran cantidad de ancho de banda, generando así un efecto visible en el consumo de CPU debido al procesamiento.

4.2.2. Paquetes perdidos

Las Figuras 4.12 4.13 4.14 y 4.15 muestran los resultados de paquetes perdidos para los 4 escenarios de unidades de CPU iniciales para cumplir con el requisito establecido de menos del 1 % en el 5G CN, los resultados se muestran en la Tabla 4.4.

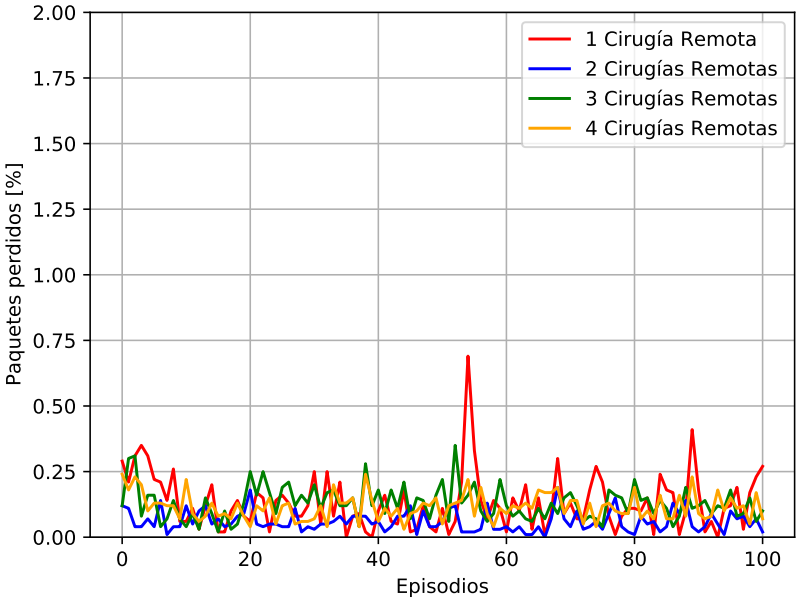

Figura 4.12: Paquetes perdidos con 1 unidad de CPU inicial

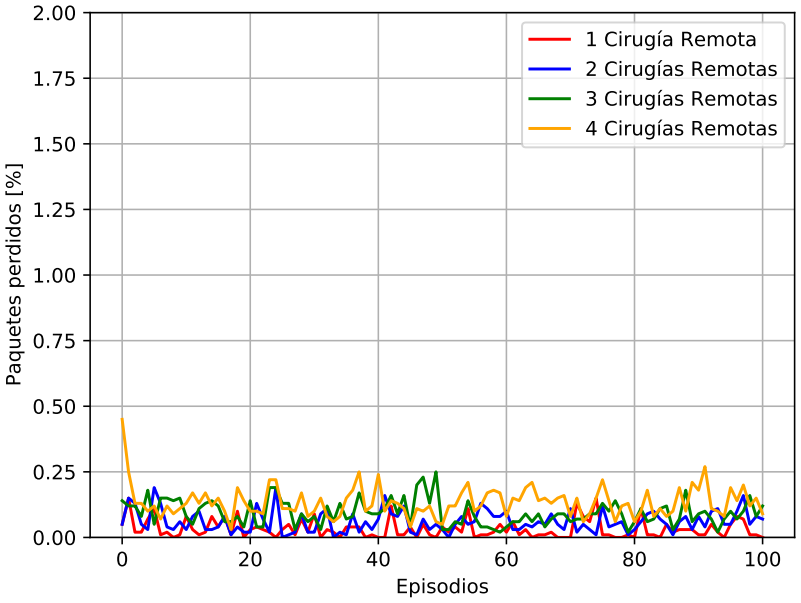

Figura 4.13: Paquetes perdidos con 3 unidades de CPU iniciales

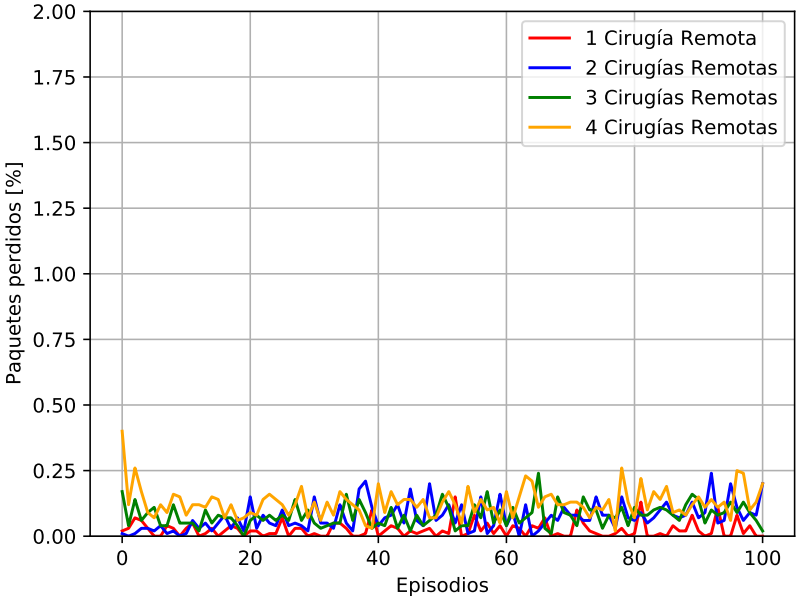

Figura 4.14: Paquetes perdidos con 5 unidades de CPU iniciales

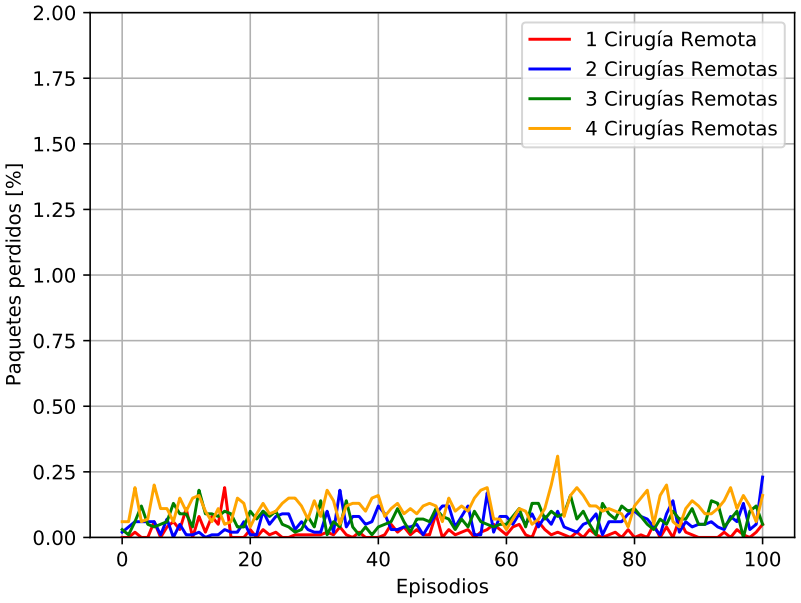

Figura 4.15: Paquetes perdidos con 7 unidades de CPU iniciales

En el caso de 1 unidad de CPU inicial (ver Figura 4.12) la tasa de pérdida de paquetes promedio es de 0.1086 %, un 90 % por debajo del umbral establecido. De manera similar, en el caso de 3 unidades de CPU iniciales (ver Figura 4.13) la tasa de pérdida de paquetes promedio es de 0.08 %, un 92 % por debajo del umbral establecido. Por otra parte, en el caso de 5 unidades de CPU iniciales (ver Figura 4.14) la tasa de pérdida de paquetes promedio es de 0.07 %, un 93 % por debajo del umbral establecido. Finalmente para 7 unidades de CPU iniciales (ver Figura 4.15) la tasa de pérdida de paquetes promedio es de 0.0665 %, un 94 % por debajo del umbral establecido.

Estos resultados evidencian el hecho de que la tasa de pérdida de paquetes no sobrepasa del 1 %. Una de las causas principales para la ocurrencia de pérdida de paquetes es la existencia de demasiados flujos conectados al mismo tiempo y haciendo un uso excesivo de la conexión. Al existir flujo de tráfico con altas tasas de transmisión, como lo son los flujos de vídeo existe la probabilidad de pérdida de paquetes, que puede afectar a la recepción de otros tipos de flujo, específicamente los hápticos,

Unidades de CPU (UPF)	Paquetes perdidos promedio
1	0.1086 %
3	0.08 %
5	0.07 %
7	0.0665 %

Tabla 4.4: Resultados paquetes perdidos

cuya tasa de transmisión es mucho más baja. Esto podría generar un cuello de botella que afecte al rendimiento general del servicio desplegado consumiendo además una mayor cantidad de recursos de CPU. Sin embargo, el mecanismo prioriza la no tolerancia de pérdida de paquetes estabilizándola debajo del 1 % requerido.

4.2.3. *Jitter*

Las Figuras 4.16 4.17 4.18 y 4.19 muestran los resultados de *Jitter* para los 4 escenarios de unidades de CPU iniciales para cumplir con el requisito establecido de menos de 30 milisegundos en el 5G CN, los resultados se muestran en la Tabla 4.5.

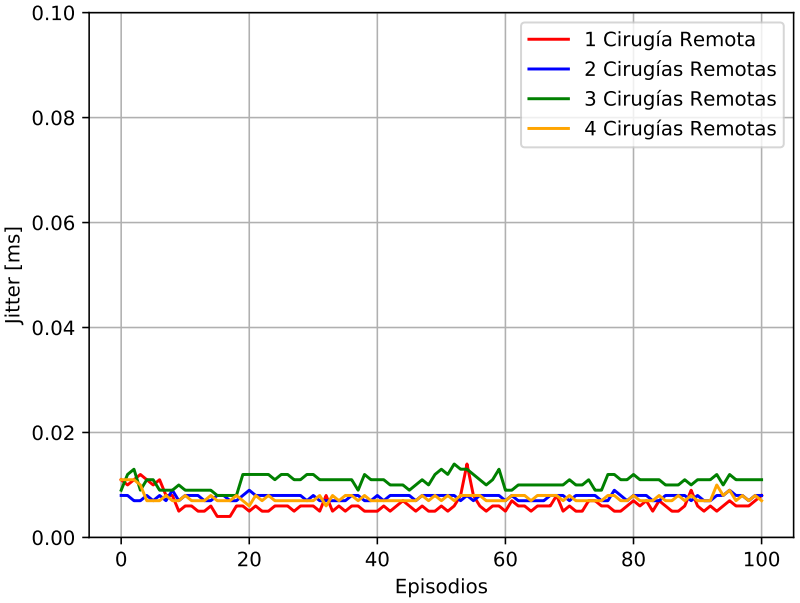

Figura 4.16: *Jitter* con 1 unidad de CPU inicial

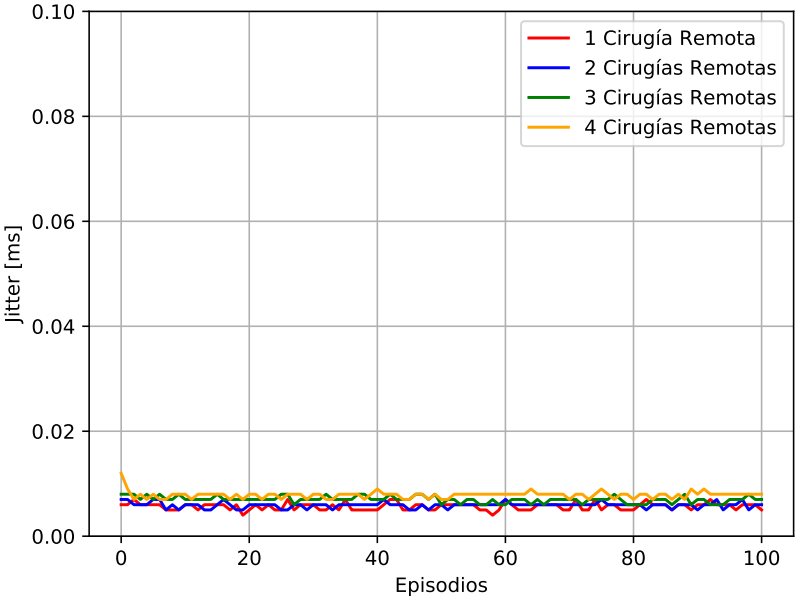

Figura 4.17: *Jitter* con 3 unidades de CPU iniciales

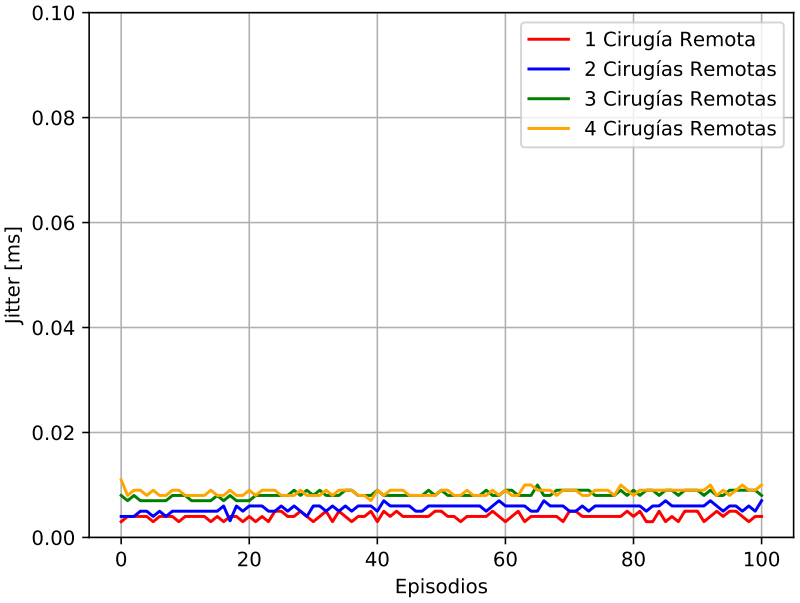

Figura 4.18: *Jitter* con 5 unidades de CPU iniciales

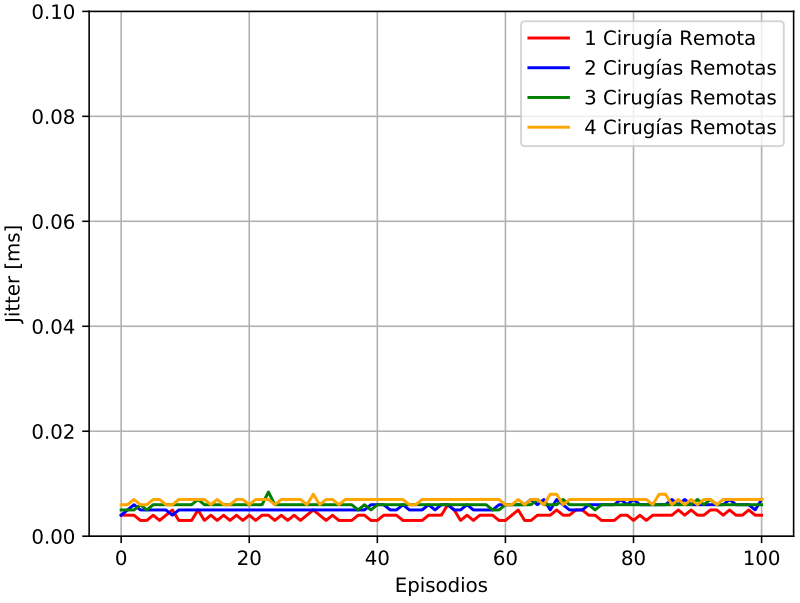

Figura 4.19: *Jitter* con 7 unidades de CPU iniciales

En el caso de 1 unidad de CPU inicial (ver Figura 4.16) el *jitter* promedio es de 0.080 milisegundos, un 99.73 % por debajo del umbral establecido. De manera similar, en el caso de 3 unidades de CPU iniciales (ver Figura 4.17) el *jitter* promedio es de 0.06 milisegundos, un 99.8 % por debajo del umbral establecido. Por otra parte, en el caso de 5 unidades de CPU iniciales (ver Figura 4.18) el *jitter* promedio es de 0.006 milisegundos, un 99.98 % por debajo del umbral establecido. Finalmente para 7 unidades de CPU iniciales (ver Figura 4.19) el *jitter* promedio es de 0.005 milisegundos, un 99.98 % por debajo del umbral establecido.

Unidades de CPU (UPF)	***Jitter* promedio [ms]**
1	0.080
3	0.06
5	0.006
7	0.005

Tabla 4.5: Resultados *Jitter*

Estos resultados evidencian el hecho de que el *jitter* no supera el valor límite de 30 milisegundos establecido. Es importante que el *jitter* no sufra de variaciones considerables debido a que el rendimiento en servicios de vídeo y VoIP se vería afectado negativamente, haciendo que el tráfico VoIP se entrecortara, o incluso se interrumpiera por completo. En el caso de uso de la cirugía remota, se tiene poca tolerancia a un *jitter* elevado debido a que puede empeorar la calidad del servicio entrecortando señales de audio o vídeo importantes para el entorno del cirujano remoto, quien manipula el sistema quirúrgico robótico basado en dicha retroalimentación visual.

Capítulo 5

Conclusiones y Trabajos Futuros

5.1. Conclusiones

En este trabajo se presenta la respuesta a la pregunta: **¿Cómo abordar la congestión de la red en un *slice* de cirugía remota en el núcleo de red 5G para satisfacer el requisito de latencia?**.

Para responder dicha pregunta se diseña un Mecanismo de Control de Congestión de un *slice* en el 5G CN, el cual se modela para emular el servicio de cirugía remota. Además, se implementa un prototipo software mediante el lenguaje de programación Python. Finalmente, una vez se implementa, se procede a realizar el análisis de rendimiento en un ambiente emulado, con el fin de observar el impacto del mecanismo en la latencia, el *jitter*, y la pérdida de paquetes. Los resultados muestran lo siguiente:

- El mecanismo de control de congestión independientemente del número de cirugías remotas logra estabilizar la latencia en un 40 % por debajo del umbral de latencia establecido para el servicio URLLC.
- Al aumentar el número de cirugías remotas, el mecanismo de control de congestión logra cumplir con los requisitos QoS de *jitter* y tasa de pérdida de paquetes, estabilizándolos en un valor 90 % debajo de los requisitos establecidos.

- La asignación de recursos para controlar la congestión en el 5G CN de un *slice* logra mitigar la latencia, reducir la tasa de pérdida de paquetes, y estabilizar el *jitter* satisfaciendo de esta manera los requisitos de QoS en un servicio URLLC como la cirugía remota.

En general, el mecanismo de control de congestión logra asignar de manera más eficiente los recursos del 5G CN. Además, el mecanismo logró cumplir con los requisitos de QoS para un *slice* en el 5G CN en términos de latencia, pérdida de paquetes y *jitter* en un ambiente de emulación.

5.2. Trabajos Futuros

- Implementar un mecanismo de control de congestión en el 5G CN enfocado en cirugía remota teniendo en cuenta mas de una métrica de recursos al mismo tiempo, tales como: CPU, memoria o GPU
- Implementar un mecanismo de control de congestión en el 5G CN enfocado en cirugía remota contemplando la disponibilidad de recursos de los *slices* adyacentes.
- Implementar un mecanismo de control de congestión en el 5G CN enfocado en cirugía remota en un entorno real.

5.3. Comentarios Finales

Se realizaron pruebas con diferentes entornos de emulación del 5G CN. Implementaciones tales como NS3, OpenSourceMano, Open5GS, Free5GC, OpenAirSim, Microk8s, UERANSIM, NetSim, y soluciones basadas en Kubernetes fueron investigadas, analizadas y testeadas con el fin de analizar el comportamiento de algunos componentes de la red 5G. Sin embargo, debido a su alta complejidad, poca flexibilidad, documentación pobre o alta demanda de recursos hardware fueron descartadas.

Bibliografía

[1] G. P. Fettweis, "The Tactile Internet: Applications and Challenges," *IEEE Vehicular Technology Magazine*, vol. 9, no. 1, pp. 64–70, Mar. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6755599/

[2] S. R. Smoot and N. K. Tan, "Branch Consolidation and WAN Optimization," in *Private Cloud Computing*. Elsevier, 2012, pp. 99–125. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780123849199000040

[3] G. Fettweis, H. Boche, T. Wiegand, and E. Zielinski, "The Tactile Internet," *ITU-T Technology Watch*, p. 24, 2014.

[4] K. S. Kim, D. K. Kim, C.-B. Chae, S. Choi, Y.-C. Ko, J. Kim, Y.-G. Lim, M. Yang, S. Kim, B. Lim, K. Lee, and K. L. Ryu, "Ultrareliable and Low-Latency Communication Techniques for Tactile Internet Services," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 376–393, Feb. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8474959/

[5] M. Aazam, K. A. Harras, and S. Zeadally, "Fog Computing for 5g Tactile Industrial Internet of Things: QoE-Aware Resource Allocation Model," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3085–3092, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8657720/

[6] C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian, "Remote surgery case: robot-assisted teleneurosurgery," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 1, 2004, pp. 819–823 Vol.1.

[7] A. Lacy, R. Bravo, A. Otero-Piñeiro, R. Pena, F. De Lacy, R. Menchaca, and J. Balibrea, "5g-assisted telementored surgery," *British Journal of Surgery*, vol. 106, no. 12, pp. 1576–1579, 2019.

[8] Q. Zhang, J. Liu, and G. Zhao, "Towards 5g Enabled Tactile Robotic Telesurgery," *arXiv:1803.03586 [cs]*, Mar. 2018, arXiv: 1803.03586. [Online]. Available: http://arxiv.org/abs/1803.03586

[9] J. S. Ladoiye, D. S. Necsulescu, and J. Sasiadek, "Kinematic Predictive Imaging Technique for Telerobotic Surgery with Time Delay using Model Predictive Control," in *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*. Międzyzdroje, Poland: IEEE, Aug. 2019, pp. 646–651. [Online]. Available: https://ieeexplore.ieee.org/document/8864678/

[10] A. Aijaz, M. Simsek, M. Dohler, and G. Fettweis, "Shaping 5g for the Tactile Internet," in *5G Mobile Communications*, W. Xiang, K. Zheng, and X. Shen, Eds. Cham: Springer International Publishing, 2017, pp. 677–691. [Online]. Available: http://link.springer.com/10.1007/978-3-319-34208-5_25

[11] H. S. Varsha and K. P. Shashikala, "The tactile Internet," in *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*. Bengaluru, India: IEEE, Feb. 2017, pp. 419–422. [Online]. Available: http://ieeexplore.ieee.org/document/7975649/

[12] X. Jiang, H. Shokri-Ghadikolaei, G. Fodor, E. Modiano, Z. Pang, M. Zorzi, and C. Fischione, "Low-Latency Networking: Where Latency Lurks and How to Tame It," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 280–306, Feb. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8452158/

[13] Y. Hao, "Investigation and technological comparison of 4g and 5g networks," *Journal of Computer and Communications*, vol. 9, pp. 36–43, 2021.

[14] N. Corporation, "Nokia Low-Latency in 4.9 and 5G Networks White Paper," Tech. Rep., 2017. [Online]. Available: https://onestore.nokia.com/asset/201407

[15] W. Kellerer, A. Basta, P. Babarczi, A. Blenk, M. He, M. Klugel, and A. M. Alba, "How to Measure Network Flexibility? A Proposal for Evaluating Softwarized Networks," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 186–192, Oct. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8284059/

[16] A. Aijaz, "A Radio Resource Slicing Framework for 5g Networks With Haptic Communications," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2285–2296, Sep. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/7831356/

[17] K. Sparks, M. Sirbu, J. Nasielski, L. Merrill, K. Leddy, P. Krishnaswamy, W. Johnston, R. Gyurek, B. Daly, M. Bayliss, J. Barnhill, and K. Balachandran, "5g network slicing whitepaper," p. 34.

[18] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8320765/

[19] "5g Network Slicing for Vertical Industries," *Global mobile Suppliers Association*, p. 17, Sep. 2017.

[20] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/6994333/

[21] W. Li, Y. Zi, L. Feng, F. Zhou, P. Yu, and X. Qiu, "Latency-Optimal Virtual Network Functions Resource Allocation for 5g Backhaul Transport Network Slicing," *Applied Sciences*, vol. 9, no. 4, p. 701, Feb. 2019. [Online]. Available: http://www.mdpi.com/2076-3417/9/4/701

[22] M. A. Imran, Y. Abdulrahman Sambo, and Q. H. Abbasi, *Enabling 5G Communication Systems to Support Vertical Industries*, 1st ed. Wiley, Aug. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119515579

[23] L. Tshiningayamwe, G.-A. Lusilao-Zodi, and M. E. Dlodlo, "A Priority Rate-Based Routing Protocol for Wireless Multimedia Sensor Networks," in *Advances in Nature and Biologically Inspired Computing*, N. Pillay, A. P. Engelbrecht, A. Abraham, M. C. du Plessis, V. Snášel, and A. K. Muda, Eds. Cham: Springer International Publishing, 2016, vol. 419, pp. 347–358. [Online]. Available: http://link.springer.com/10.1007/978-3-319-27400-3_31

[24] W. Chen, Y. Niu, and Y. Zou, "Congestion control and energy-balanced scheme based on the hierarchy for WSNs," *IET Wireless Sensor Systems*, vol. 7, no. 1, pp. 1–8, Feb. 2017. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-wss.2015.0097

[25] T. T. Zin, J. C.-W. Lin, J.-S. Pan, P. Tin, and M. Y. (eds.), *Genetic and Evolutionary Computing: Proceedings of the Ninth International Conference on Genetic and Evolutionary Computing, August 26-28, 2015, Yangon, Myanmar - Volume II*, 1st ed., ser. Advances in Intelligent Systems and Computing 388. Springer International Publishing, 2016. [Online]. Available: http://gen.lib.rus.ec/book/index.php?md5=94946c57d5bb7093eed5368993e2dd87%7D

[26] H. Al-Bahadili, *Simulation in Computer Network Design and Modeling: Use and Analysis: Use and Analysis*, ser. Premier reference source. Information Science Reference, 2012. [Online]. Available: https://books.google.com.co/books?id=uNlplf2C03QC

[27] Y.-J. Chen, L.-Y. Cheng, and L.-C. Wang, "Prioritized resource reservation for reducing random access delay in 5g URLLC," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Montreal, QC: IEEE, Oct. 2017, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/8292695/

[28] B. Wang and J. Su, "A survey of elephant flow detection in SDN," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*. Antalya, Turkey: IEEE, Mar. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8355352

[29] X. Li, J. Yan, and H. Ren, "Software Defined Traffic Engineering for Improving Quality of Service," *China Communications*, p. 14, 2017.

[30] Liang Guo and I. Matta, "The war between mice and elephants," in *Proceedings Ninth International Conference on Network Protocols. ICNP 2001.* Riverside, CA, USA: IEEE Comput. Soc, 2001, pp. 180–188. [Online]. Available: http://ieeexplore.ieee.org/document/992898/

[31] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2432–2455, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7932863/

[32] Y. Kong, H. Zang, and X. Ma, "Improving TCP Congestion Control with Machine Intelligence," in *Proceedings of the 2018 Workshop on Network Meets AI & ML - NetAI'18.* Budapest, Hungary: ACM Press, 2018, pp. 60–66. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3229543.3229550

[33] N. Yuvaraj and P. Thangaraj, "Machine learning based adaptive congestion window adjustment for Congestion Aware Routing in Cross Layer Approach Handling of Wireless Mesh Network," *Cluster Computing*, vol. 22, no. S4, pp. 9929–9939, Jul. 2019. [Online]. Available: http://link.springer.com/10.1007/s10586-018-2357-y

[34] G. Kassem, I. Ahmad, F. Hameed, and A. Zakariyya, "Tcp variants: An overview," in *2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, 2010, pp. 536–540.

[35] J. Aina, L. Mhamdi, and H. Hamdi, "F-DCTCP: Fair Congestion Control for SDN-Based Data Center Networks," in *2019 International Symposium on Networks, Computers and Communications (ISNCC).* Istanbul, Turkey: IEEE, Jun. 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8909171/

[36] V. Vasudevan, A. Phanishayee, H. Shah, E. Krevat, D. G. Andersen, G. R. Ganger, G. A. Gibson, and B. Mueller, "Safe and Effective Fine-grained TCP Retransmissions for Datacenter Communication," p. 12.

[37] G. Zhu, J. Zan, Y. Yang, and X. Qi, "A Supervised Learning Based QoS Assurance Architecture for 5g Networks," *IEEE Access*, vol. 7, pp. 43 598–43 606, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8673765/

[38] N. Jay, N. Rotman, B. Godfrey, M. Schapira, and A. Tamar, "A deep reinforcement learning perspective on internet congestion control," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 3050–3059. [Online]. Available: http://proceedings.mlr.press/v97/jay19a.html

[39] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," in *Machine learning techniques for multimedia*. Springer, 2008, pp. 21–49.

[40] K. Han, A. Hwang, J. Y. Lee, and B. C. Kim, "Design and performance evaluation of enhanced congestion control algorithm for wireless TCP by using a deep learning," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*. Honolulu, HI: IEEE, Jan. 2018, pp. 1–2. [Online]. Available: http://ieeexplore.ieee.org/document/8330648/

[41] Y. Liang, L. Jiang, C. He, D. He, and P. Li, "Performance Analyses of Quantized Congestion Notification for 5g Distributed Base Station," in *2018 3rd Cloudification of the Internet of Things (CIoT)*. Paris, France: IEEE, Jul. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8627093/

[42] B. Han, A. DeDomenico, G. Dandachi, A. Drosou, D. Tzovaras, R. Querio, F. Moggio, O. Bulakci, and H. D. Schotten, "Admission and Congestion Control for 5g Network Slicing," in *2018 IEEE Conference on Standards for*

Communications and Networking (CSCN). Paris, France: IEEE, Oct. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8581773/

[43] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE access*, vol. 6, pp. 3619–3647, 2017.

[44] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, "5g wireless network slicing for embb, urllc, and mmtc: A communication-theoretic view," *Ieee Access*, vol. 6, pp. 55 765–55 779, 2018.

[45] A. Americas 5G, "Network Slicing for 5g Networks & Services," 5G Americas White Paper, Nov. 2016. [Online]. Available: https://www.5gamericas.org/wp-content/uploads/2019/07/5G_Americas_Network_Slicing_11.21_Final.pdf

[46] B. Blanco, J. O. Fajardo, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-Romero, I. Trajkovska, P. S. Khodashenas, L. Goratti, M. Paolino *et al.*, "Technology pillars in the architecture of future 5g mobile networks: Nfv, mec and sdn," *Computer Standards & Interfaces*, vol. 54, pp. 216–228, 2017.

[47] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. Leung, "Network slicing based 5g and future mobile networks: mobility, resource management, and challenges," *IEEE communications magazine*, vol. 55, no. 8, pp. 138–145, 2017.

[48] M. Berman, J. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "Geni: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, 03 2014.

[49] NGMN Alliance, "Description of Network Slicing Concept White Paper," Sep. 2016.

[50] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5g: Challenges and opportunities," *IEEE Internet Computing*, vol. 21, no. 5, pp. 20–27, 2017.

[51] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5g with sdn/nfv: Concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, May 2017.

[52] T. Shimojo, M. R. Sama, A. Khan, and S. Iwashina, "Cost-efficient method for managing network slices in a multi-service 5g core network," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, May 2017, pp. 1121–1126.

[53] M.-K. Shin, K.-H. Nam, and H.-J. Kim, "Software-defined networking (sdn): A reference architecture and open apis," in *2012 International Conference on ICT Convergence (ICTC)*. IEEE, 2012, pp. 360–361.

[54] C. Peliza, F. Dufour, A. Serra, G. Micieli, and F. Guerrero, "Virtualización de funciones de red," in *XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan).*, 2019.

[55] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 236–262, 2015.

[56] E. Ganesan, I.-S. Hwang, A. T. Liem, and M. S. Ab-Rahman, "5g-enabled tactile internet resource provision via software-defined optical access networks (sdoans)," *Photonics*, vol. 8, no. 5, 2021. [Online]. Available: https://www.mdpi.com/2304-6732/8/5/140

[57] A. Rovetta, R. Sala, Xia Wen, and A. Togno, "Remote control in telerobotic surgery," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 26, no. 4, pp. 438–444, Jul. 1996. [Online]. Available: http://ieeexplore.ieee.org/document/508822/

[58] G. Ballantyne, "Robotic surgery, telerobotic surgery, telepresence, and telementoring," *Surgical Endoscopy*, vol. 16, no. 10, pp. 1389–1402, Oct. 2002. [Online]. Available: http://link.springer.com/10.1007/s00464-001-8283-7

[59] M. Wazid, A. K. Das, and J.-H. Lee, "User authentication in a tactile internet based remote surgery environment: Security issues, challenges, and future research directions," *Pervasive and Mobile Computing*, vol. 54, pp. 71–85, Mar. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1574119218304784

[60] A. M. Okamura, "Haptic feedback in robot-assisted minimally invasive surgery," *Current opinion in urology*, vol. 19, no. 1, pp. 102–107, Jan 2009, 19057225[pmid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/19057225

[61] M. Simsek, A. Aijaz, M. Dohler, J. Sachs, and G. Fettweis, "5g-Enabled Tactile Internet," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 460–473, Mar. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7403840/

[62] Y. Fu, S. Wang, C.-X. Wang, X. Hong, and S. McLaughlin, "Artificial Intelligence to Manage Network Traffic of 5g Wireless Networks," *IEEE Network*, vol. 32, no. 6, pp. 58–64, Nov. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8553655/

[63] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[64] G. E. Monahan, "State of the art—a survey of partially observable markov decision processes: theory, models, and algorithms," *Management science*, vol. 28, no. 1, pp. 1–16, 1982.

[65] L. S. Shapley, "Stochastic games," *Proceedings of the national academy of sciences*, vol. 39, no. 10, pp. 1095–1100, 1953.

[66] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[67] S. Haeri, M. Arianezhad, and L. Trajkovic, "A predictive q-learning algorithm for deflection routing in buffer-less networks," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 764–769.

[68] J. Yuan, X. Jiang, L. Zhong, and H. Yu, "Energy aware resource scheduling algorithm for data center using reinforcement learning," in *2012 Fifth International Conference on Intelligent Computation Technology and Automation*, 2012, pp. 435–438.

[69] A. S. Randrianasolo and L. D. Pyeatt, "Q-learning: From computer network security to software security," in *2014 13th International Conference on Machine Learning and Applications*, 2014, pp. 257–262.

[70] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[71] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, 1994.

[72] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[73] P. Sun, J. Li, J. Lan, Y. Hu, and X. Lu, "Rnn deep reinforcement learning for routing optimization," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, 2018, pp. 285–289.

[74] A. Jeerige, D. Bein, and A. Verma, "Comparison of deep reinforcement learning approaches for intelligent game playing," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0366–0371.

[75] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[76] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[77] Y. Lin, X. Dai, L. Li, and F.-Y. Wang, "An efficient deep reinforcement learning model for urban traffic control," *arXiv preprint arXiv:1808.01876*, 2018.

[78] T. Li, X. Zhu, and X. Liu, "An end-to-end network slicing algorithm based on deep q-learning for 5g network," *IEEE Access*, vol. 8, pp. 122 229–122 240, 2020.

[79] L. Wang, W. Mao, J. Zhao, and Y. Xu, "Ddqp: A double deep q-learning approach to online fault-tolerant sfc placement," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 118–132, 2021.

[80] X. Tao and A. S. Hafid, "Deepsensing: A novel mobile crowdsensing framework with double deep q-network and prioritized experience replay," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 547–11 558, 2020.

[81] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial internet of things: A dueling deep Q -learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4627–4639, 2019.

[82] X. Tan, Y. Lee, C.-B. Chng, K.-B. Lim, and C.-K. Chui, "Robot-assisted flexible needle insertion using universal distributional deep reinforcement learning," *International journal of computer assisted radiology and surgery*, vol. 15, no. 2, p. 341—349, February 2020. [Online]. Available: https://doi.org/10.1007/s11548-019-02098-7

[83] V. Jacobson, "Congestion Avoidance and Control," vol. 18, p. 17, Aug. 1988. [Online]. Available: http://www.cs.binghamton.edu/~nael/cs428-528/deeper/jacobson-congestion.pdf

[84] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and Van Jacobson, "BBR: congestion-based congestion control," *Communications of the ACM*, vol. 60, no. 2, pp. 58–66, Jan. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3042068.3009824

[85] D. Scholz, B. Jaeger, L. Schwaighofer, D. Raumer, F. Geyer, and G. Carle, "Towards a Deeper Understanding of TCP BBR Congestion Control," in *2018 IFIP Networking Conference (IFIP Networking) and Workshops*. Zurich, Switzerland: IEEE, May 2018, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/8696830/

[86] N. Dukkipati and N. McKeown, "Why Flow-Completion Time is the Right metric for Congestion Control and why this means we need new algorithms," p. 8.

[87] I. A. Najm, M. Ismail, and G. A. Abed, "High-Performance Mobile Technology LTE-A using the Stream Control Transmission Protocol: A Systematic Review and Hands-on Analysis," *Journal of Applied Sciences*, vol. 14, no. 19, pp. 2194–2218, Dec. 2014. [Online]. Available: http://www.scialert.net/abstract/?doi=jas.2014.2194.2218

[88] I. A. Najm, M. Ismail, J. Lloret, K. Z. Ghafoor, B. Zaidan, and A. A.-r. T. Rahem, "Improvement of SCTP congestion control in the LTE-A network," *Journal of Network and Computer Applications*, vol. 58, pp. 119–129, Dec. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S108480451500209X

[89] I. Abdeljaouad, H. Rachidi, S. Fernandes, and A. Karmouch, "Performance analysis of modern tcp variants: A comparison of cubic, compound and new reno," in *2010 25th Biennial Symposium on Communications*, 2010, pp. 80–83.

[90] W. Li, F. Zhou, K. R. Chowdhury, and W. Meleis, "Qtcp: Adaptive congestion control with reinforcement learning," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 3, pp. 445–458, 2019.

[91] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. S. Wong, "D-ACB: Adaptive Congestion Control Algorithm for Bursty M2m Traffic in LTE Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9847–9861, Dec. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7404058/

[92] S. K. Kasera, R. Ramjee, S. R. Thuel, and X. Wang, "Congestion Control Policies for IP-Based CDMA Radio Access Networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, p. 14, 2005.

[93] Chung-Ju Chang, Bo-Wei Chen, Terng-Yuan Liu, and Fang-Ching Ren, "Fuzzy/neural congestion control for integrated voice and data DS-CDMA/FRMA cellular networks," *IEEE Journal on Selected Areas in*

Communications, vol. 18, no. 2, pp. 283–293, Feb. 2000. [Online]. Available: http://ieeexplore.ieee.org/document/824814/

[94] G. Pocovi, K. I. Pedersen, B. Soret, M. Lauridsen, and P. Mogensen, "On the impact of multi-user traffic dynamics on low latency communications," in *2016 International Symposium on Wireless Communication Systems (ISWCS)*. Poznan, Poland: IEEE, Sep. 2016, pp. 204–208. [Online]. Available: http://ieeexplore.ieee.org/document/7600901/

[95] J. Zhang, X. Xu, K. Zhang, B. Zhang, X. Tao, and P. Zhang, "Machine Learning Based Flexible Transmission Time Interval Scheduling for eMBB and uRLLC Coexistence Scenario," *IEEE Access*, vol. 7, pp. 65 811–65 820, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8718287/

[96] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "Tcp westwood: End-to-end congestion control for wired/wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 467–479, Sep 2002. [Online]. Available: https://doi.org/10.1023/A:1016590112381

[97] C. P. Fu and S. C. Liew, "Tcp veno: Tcp enhancement for transmission over wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 2, pp. 216–228, 2003. [Online]. Available: https://doi.org/10.1109/JSAC.2002.807336

[98] M. Nasimi, M. A. Habibi, B. Han, and H. D. Schotten, "Edge-Assisted Congestion Control Mechanism for 5g Network Using Software-Defined Networking," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. Lisbon: IEEE, Aug. 2018, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8491233/

[99] I. A. Najm, A. K. Hamoud, J. Lloret, and I. Bosch, "Machine Learning Prediction Approach to Enhance Congestion Control in 5g IoT Environment," *Electronics*, vol. 8, no. 6, p. 607, May 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/6/607

[100] Y. Han, M. Zuo, H. Yuan, Y. Zhong, Z. Yuan, and T. Bi, "A qos-based fairness-aware bbr congestion control algorithm using quic," *Wireless*

Communications and Mobile Computing, vol. 2022, p. 7222030, Apr 2022. [Online]. Available: https://doi.org/10.1155/2022/7222030

[101] X. Xiao, M. Zhao, and Y. Zhu, "Multi-stage resource-aware congestion control algorithm in edge computing environment," *Energy Reports*, vol. 8, pp. 6321–6331, Nov 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484722008423

[102] G.-H. Kim and Y.-Z. Cho, "mms-tcp: Scalable tcp for improving throughput and fairness in 5g mmwave networks," 2022. [Online]. Available: https://doi.org/10.3390/s22103609

[103] R. Morris, "Scalable tcp congestion control," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, ser. Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064), vol. 3, 2000, pp. 1176–1183 vol.3. [Online]. Available: https://doi.org/10.1109/INFCOM.2000.832487

[104] Y. Song, H. Ni, and X. Zhu, "Two-level congestion control mechanism (2lccm) for information-centric networking," 2021. [Online]. Available: https://doi.org/10.3390/fi13060149

[105] N. F. V. NFV, "Etsi gs nfv-sec 001 v1. 1.1 (2014-10)," 2014.

[106] W. Stallings, *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Pearson Education, 2015. [Online]. Available: https://books.google.es/books?id=nL_QCgAAQBAJ

[107] G. Brown, "Service-based architecture for 5g core networks," pp. 1–12, 2017. [Online]. Available: https://www.huawei.com/en/press-events/news/2017/11/HeavyReading-WhitePaper-5G-Core-Network

[108] B. Chatras, U. S. Tsang Kwong, and N. Bihannic, "Nfv enabling network slicing for 5g," in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, 2017, pp. 219–225.

[109] CISCO, "Amf overview •," pp. 1–8. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/wireless/ucc/amf/2021-04/config-and-admin/b_ucc-5g-amf-config-and-admin-guide_2021-04/m_amf-overview.pdf#page=1&zoom=auto,-190,492

[110] ——, "5g smf overview," pp. 1–16. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/wireless/ucc/smf/b_SMF/b_SMF_chapter_010010.pdf

[111] ——, "5g-upf overview," pp. 1–10. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/wireless/ucc/upf/2020-03/b_ucc-5g-upf-config-and-admin-guide_2020-03/b_UPF_chapter_011011.pdf

[112] D. M. Casas-Velasco, O. M. C. Rendon, and N. L. da Fonseca, "Intelligent routing based on reinforcement learning for software-defined networking," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 870–881, 2020.

Mecanismo de control de congestión para la latencia en un network slice dedicado a URLLC: un caso de estudio en cirugía remota

ANEXOS

Trabajo de pregrado

Kevin Muñoz Rengifo
Juan Manuel Solís Prado

Advisor: PhD. Oscar Mauricio Caicedo Rendón
Co-Advisor: Msc. Johanna Andrea Hurtado

Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2022

Anexo A

Algoritmos

El Anexo A presenta el enlace de redireccionamiento al repositorio en GitHub, en el cual están los respectivos códigos fuente del Mecanismo de Control de Congestión realizado.

`https://github.com/kevinmuz55/Congestion-Control-DRL-Algorithm`

Anexo B

Publicaciones

El Anexo B presenta el artículo enviado a la comunidad científica con el propósito de que sea publicado.

- **Kevin Muñoz Rengifo**, **Juan Manuel Solis Prado**, Oscar Mauricio Caicedo Rendón, Johanna Andrea Hurtado Sanchez. **Congestion Control Mechanism for latency in a 5G Network Slice dedicated to URLLC: a case study in remote surgery**. IEEE Latincom.
    - Estado: Escrito y por enviar.
    - Clasificación: B.
    - Factor de Impacto: 0.96
    - Índice H: 4 Scimago