

**Control de admisión basado en aprendizaje por
refuerzo profundo para el servicio de cirugía
remota en 5G**



Trabajo de Grado

Santiago de Jesus Martinez Semanate

Jose Luis Rincón Zapata

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

*Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2021*

**Control de admisión basado en aprendizaje por
refuerzo profundo para el servicio de cirugía
remota en 5G**

Santiago de Jesus Martinez Semanate
Jose Luis Rincón Zapata

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: M.sc. Johanna Andrea Hurtado Sanchez

*Departamento de Telemática
Facultad de Ingeniería Electrónica y Telecomunicaciones
Universidad del Cauca
Popayán, Cauca, 2021*

AGRADECIMIENTOS

En primera medida es imperativo agradecer a Dios, a nuestras familias, especialmente a nuestros padres, madres y hermanos que fueron un apoyo en todos los momentos vividos de este proceso. Agradecer a todas las personas que hicieron parte de nuestra vida universitaria, incluyendo amigos, compañeros y ahora colegas del programa de Ingeniería Electrónica y Telecomunicaciones, los cuales fueron un pilar y apoyo importante en toda nuestra etapa formativa. Agradecer enormemente a nuestro director Oscar Mauricio Caicedo, quien se convirtió en un guía, consejero y gran mentor académico, personal y profesional. De la misma manera, agradecemos a nuestra codirectora Johanna Andrea Hurtado, quien desempeñó un papel importante en el desarrollo de este trabajo de grado. Mencionar además, al semillero de investigación COMSOCAUCA y en especial a William Fernando Villota quien nos brindó las bases teóricas para 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 perduraran por toda nuestra vida, conocer profesionales con formaciones y valores humanos excelentes, convirtiéndose así en el lugar donde logramos cumplir metas y sueños.

Resumen

Los *slices* de red para el servicio de cirugía remota representan un desafío importante para las redes 5G dados sus estrictos requisitos en términos de confiabilidad y latencia. Múltiples solicitudes de *slices* de red para cirugía remota llegarán a un proveedor de infraestructura que deberá atenderlas y satisfacerlas eficientemente. Un mecanismo de Control de Admisión de *slices* de red puede contribuir al cumplimiento de los requisitos impuestos por el servicio de cirugía remota. La literatura revela que existen propuestas basadas en Teoría de Colas, Heurísticas, aprendizaje por refuerzo y aprendizaje por refuerzo profundo, sin embargo, estas propuestas no consideran la satisfacción de la latencia en los *slices* de red para la cirugía remota y la optimización del lucro, como una tarea conjunta. Este trabajo de grado propone un mecanismo de control de admisión enfocado en cirugía remota, para satisfacer los altos requerimientos de latencia de este servicio y la optimización del lucro de los proveedores de infraestructura. Este mecanismo es simulado en topologías de red con 32 y 64 nodos basadas en Puntos de Presencia de la Infraestructura de virtualización de funciones de red y evaluado mediante el lucro y la tasa de satisfacción de latencia. Los resultados muestran que este mecanismo logra un equilibrio adecuado entre la tasa de satisfacción y la optimización del lucro, demostrando ser una solución viable para un control de admisión de *slices* de red enfocado en cirugía remota.

Contenido

Lista de Figuras	VI
Lista de Tablas	VII
Lista de Acrónimos	VIII
1 INTRODUCCIÓN	1
1.1 Definición del Problema	1
1.2 Objetivos	5
1.2.1 Objetivo General	5
1.2.2 Objetivos Específicos	5
1.3 Aportes y Envío	6
1.3.1 Aportes	6
1.3.2 Envío	6
1.4 Estructura del Documento	7
2 CONCEPTOS FUNDAMENTALES	8
2.1 Redes de Quinta Generación	8

CONTENIDO

2.2	Redes Lógicas	9
2.3	Cirugía Remota	11
2.4	Control de Admisión	12
2.5	Aprendizaje por Refuerzo Profundo	14
3	TRABAJOS RELACIONADOS	17
3.1	Control de Admisión con Algoritmos Heurísticos	17
3.2	Control de Admisión Basado en Aprendizaje por Refuerzo	19
3.3	Control de Admisión Basado en Aprendizaje por Refuerzo Profundo	21
3.4	Brechas	22
4	Control de Admisión basado en Aprendizaje por Refuerzo Profundo para Cirugía Remota en 5G	24
4.1	Arquitectura	24
4.1.1	Motivación	24
4.1.2	Visión General	26
4.1.3	Módulos	26
4.2	Agente Basado en Aprendizaje por Refuerzo Profundo para el Control de Admisión del Servicio de Cirugía Remota en 5G	32
4.2.1	Espacio de Estados	33
4.2.2	Espacio de Acciones	34
4.2.3	Función de Recompensa	34
4.2.4	Redes Neuronales	39

CONTENIDO

4.2.5	Reproducción de Experiencia	40
4.2.6	Exploración y Explotación	40
4.3	Algoritmo de Control de Admisión Basado en Aprendizaje por Refuerzo Profundo Enfocado en el Servicio de Cirugía Remota	41
5	Evaluación y Análisis de Resultados	45
5.1	Evaluación	45
5.1.1	Ambiente de Experimentación	45
5.1.2	Métricas	48
5.1.3	Configuración de los Experimentos	49
5.2	Resultados y Análisis	57
5.2.1	Red de 32 Nodos	57
5.2.2	Red de 64 Nodos	59
6	Conclusiones y Trabajos Futuros	61
6.1	Conclusiones	61
6.2	Trabajos futuros	62

Lista de Figuras

2.1	Tecnologías habilitadoras de 5G para una cirugía remota	12
2.2	Ejemplo de control de admisión	13
4.1	Arquitectura del Mecanismo de Control de Admisión	27
4.2	Grafos de las SLR	28
4.3	Slices para cirugía remota y otras aplicaciones	29
4.4	Arquitectura interna del agente deep reinforcement learning	38
5.1	Topologías de 32 y 64 nodos	46
5.2	Arquitectura del ambiente de experimentación	47
5.3	interacción de archivos python	48
5.4	Variación de potenciales de satisfacción	51
5.5	Variación de α	53
5.6	Variación δ_1, δ_2	54
5.7	Variación de $hl, neu, \gamma, bs, \tau$	55
5.8	Análisis de rendimiento en topología de 32 nodos	58
5.9	Rendimiento del control de admisión según la topología	60

Lista de Tablas

2.1	Requisitos latencia y ancho de banda de los casos de uso 5G	13
3.1	Trabajos relacionados	22
5.1	Requerimientos de calidad de servicio para los <i>Slices Request</i>	46
5.2	Variación de β	50
5.3	Variación de α	50
5.4	Variación de δ	52
5.5	Configuración de Parámetros	56

Lista de acrónimos

Acrónimo	Significado en inglés	Significado en español
AC	Admission Control	Control de Admisión
AMF	Access and Mobility Management Function	Función de Gestión de Acceso y Movilidad
ANN	Artificial Neural Networks	Redes Neuronales Artificiales
ATM	Asynchronous Transfer Mode	Modo de Transferencia Asíncrona
BDA	Big Data Analytics	Análisis de Big Data
CP	Control Plane	Plano de Control
CUPS	Control and User Plane Separation	Separación del Plano de Control y Usuario
DDQL	Double DQL	Doble Q-Learning Profundo
DL	Deep Learning	Aprendizaje Profundo
DNN	Deep Neural Networks	Redes Neuronales Profundas
DRL	Deep Reinforcement Learning	Aprendizaje por Refuerzo Profundo
DQL	Deep Q-Learning	Q-Learning Profundo
DQN	Deep Q-Network	Red Q profunda
eMBB	enhanced Mobile BroadBand	Banda ancha móvil mejorada
ETSI	European Telecommunications Standards Institute	Instituto Europeo de Normas de Telecomunicaciones
HW	Holt-Winters	n.a.
InfP	Infrastructure Provider	Proveedor de Infraestructura
IT	Information Technologies	Tecnologías de la Información
ITU	International Telecommunications Union	Unión Internacional de Telecomunicaciones
ITU-R	International Telecommunications Union—Radiocommunication Sector	Unión Internacional Telecomunicaciones-Sector de Radiocomunicaciones
ITU-T	International Telecommunications Union—Telecommunication Sector	Unión Internacional Telecomunicaciones-Sector de Telecomunicaciones
KP	Knapsack Problem	Problema de la mochila
KPI	Key Performance Indicator	Indicador Clave de Rendimiento
MDP	Markov Decision Process	Proceso de Decisión de Markov
mMTC	massive Machine type communications	Comunicaciones Masivas Tipo Maquina
ML	Machine Learning	Aprendizaje Automático
MNO	Mobile Network Operator	Operador de Red Móvil
MPLBAC	Algorithm for Multipath routing with Load Balancing and Admission Control	Algoritmo para enrutamiento de múltiples rutas con balanceo de carga y control de admisión
MSE	Mean Squared Error	Error Medio Cuadrático
N3AC	Network-slicing Neural Network Admission Control	Control de Admisión de Redes Neuronales de Slices de Red
NFV	Network Function Virtualization	Virtualización de funciones de red
NFVI-PoP	Network-slicing Neural Network Admission Control	Puntos de presencia de infraestructura de virtualización de funciones de red
NSL	Network Slicing	Corte de red
OvC&R	Overload Cost and Requirements	Costos y requisitos de sobrecarga
QoE	Quality of Experience	Calidad de Experiencia
QoS	Quality of Service	Calidad de Servicio
RAN	Radio Access Network	Red de Acceso Radio
RL	Reinforcement Learning	Aprendizaje por Refuerzo
RS	(Haptic Remote Surgery)	Cirugía remota háptica
SaR	Satisfaction Ratio	Tasa de Satisfacción
SDN	Software Defined Networking	Redes Definidas por Software
SLAW	Self-similar Least Action Walk	Caminata de Acción Mínima Auto-Similar
SLR	Slices Request	Solicitud de slice
SMDP	Semi-Markov Decision Process	Proceso de Decisión de Semi-Markov
SMF	Session Management Function	Función de Gestión de Sesiones
UP	User Plane	Plano de Usuario
URLLC	Ultra-reliable low-latency communications	Comunicaciones de baja latencia ultra fiables
VM	Virtual Machine	Maquina Virtual
WHO	Central World Health Organization	Organización Mundial de la Salud

Capítulo 1

INTRODUCCIÓN

1.1 Definición del Problema

La WHO (*World Health Organization*) estima que anualmente más de 2 mil millones de personas no reciben atención quirúrgica básica de manera oportuna debido a la escasez de cirujanos [1]. Esta cifra representa una gran problemática, simbolizando el incumplimiento del principio fundamental de conservación de la vida de todo ser humano. La cirugía remota, definida como un procedimiento quirúrgico realizado a distancia, donde un cirujano controla remotamente un equipo robotizado [2], pretende mitigar esta problemática permitiendo llevar el conocimiento de los mejores cirujanos a cualquier intervención quirúrgica en el mundo sin importar las barreras socio-económicas.

La cirugía remota es un servicio vertical [3, 4], enmarcado dentro del caso de uso URLLC (*Ultra-reliable low-latency communication*) [5] de las redes de quinta generación (5G); URLLC está orientado a servicios de alta confiabilidad y estrictos tiempos de latencia. Cisco define la latencia como la cantidad de tiempo que le toma a los datos recorrer la red desde un punto determinado hasta otro [6]. La ITU (*International Telecommunications Union*), junto a diversas investigaciones, establece el máximo valor de latencia para una cirugía remota en 1 ms extremo a extremo [5, 7–10], de los cuales [0.1,0.5] ms deben ser satisfechos por el núcleo de la

red [7, 11]. Este valor de latencia representa un reto para el núcleo de una red 5G, demandando una eficiente gestión de recursos físicos y virtuales.

NSL (*Network Slicing*) es un concepto utilizado en 5G para cumplir con los estrictos requisitos de QoS (*Quality of Service*) demandados por los servicios de la industria vertical [4], tal es el caso de la cirugía remota. NSL divide los recursos de la infraestructura física del InP (*Infrastructure Provider*) [12], creando redes lógicas denominadas *slices* de red, las cuales son personalizadas y desplegadas según las exigencias propias del servicio solicitado por los *tenant*¹. En el ecosistema de NSL, los InP reciben peticiones de los *tenants* para la creación de *slices*, las cuales deben ser correctamente seleccionadas para su posterior ingreso y despliegue en la red. Este proceso de selección, denominado AC (*Control Admisión*), tiene un gran nivel de importancia dado que la ineficiente aceptación de peticiones resultaría en el uso inadecuado de los recursos, el incumplimiento de las exigencias de latencia demandadas por el servicio de cirugía remota y la afectación del lucro² del InP. Por ejemplo, un mecanismo de AC podría aceptar un número elevado de peticiones, las cuales la infraestructura no lograría satisfacer o un grupo específico de peticiones que no aseguran la optimización del lucro del InP. Lograr este correcto balance entre estos dos métricas (satisfacción de latencia y lucro) aseguraría un beneficio mutuo entre el *tenant* y el InP.

Las redes actuales ofrecen mecanismos de AC limitados por el uso de arquitecturas de red rígidas y estáticas, impidiendo el uso eficiente de los recursos de red [14–17]. Por esta razón, es necesario contar con nuevos mecanismos de AC de *slices* de red, diseñados para explotar la flexibilidad presentada por las redes 5G. Estos mecanismos deben ser sensibles al contexto [9], es decir, capaces de adaptarse al estado de la red de manera inteligente y autónoma según las necesidades del servicio, considerando diversas políticas para uno o un conjunto de *slices* [13]. Ejemplos de políticas consideradas son la maximización del lucro del InP, la minimización y distribución del uso de recursos en la infraestructura de red física, y la satisfacción de la latencia en los enlaces por los cuales transitan flujos sensibles al retardo.

¹Un *tenant* hace parte del ecosistema de NSL, donde los InP permiten la creación de *slices* con requisitos propios del servicio [13].

²Ganancia que obtiene un InP.

Los enfoques [18–21] muestran como los algoritmos heurísticos pueden ser útiles para mejorar el AC de slices de red en términos de reducción de costos relacionados al uso de los recursos y satisfacción de requisitos como la velocidad, latencia, disponibilidad y fiabilidad de la red. En particular, [18] presenta un modelo de *slice* de red único y preestablecido donde muchos usuarios comparten un solo *slice*, lo cual podría perjudicar el principio de aislamiento [22] (una clave fundamental en el diseño de NSL [23–25]) y la QoS requerida por el servicio de cirugía remota. Los autores de [19], hacen uso de algoritmos reactivos (no predictivos), ignorando los estados futuros de la red, los cuales permiten obtener información más completa para la toma de decisiones. [20] considera como política de AC la maximización del lucro del InP e ignora la latencia, un elemento fundamental para el servicio de cirugía remota.

Por otra parte, RL (Aprendizaje por Refuerzo) se ha convertido en una técnica popular en recientes investigaciones, donde un agente toma decisiones periódicas basadas en la interacción con el entorno y ajusta automáticamente su estrategia para lograr políticas óptimas dirigidas a un objetivo específico [26]. Propuestas como [13, 27, 28] desarrollan modelos de AC de *slices* de red basados en técnicas de aprendizaje por refuerzo, las cuales aprovechan la flexibilidad de las redes 5G. Su principal objetivo es maximizar el lucro, mediante un AC eficiente que optimice el uso de los limitados recursos de red. No obstante, [27, 28] presentan problemas de escalabilidad, donde al aumentar significativamente el espacio de estados y acciones del MDP³ (*Markov Decision Process*), la complejidad computacional se torna inmanejable. Además, ninguna de estas propuestas contribuye con la satisfacción de los requerimientos presentados por el servicio de cirugía remota. En el caso de [13], aunque presenta un exhaustivo análisis del AC de *slices* de red, todos sus esfuerzos se centran en la gestión de los recursos de la RAN (*Radio Access Network*), ignorando las latencias presentes en el núcleo de la red, los cuales son abordados en este trabajo de grado. Recientemente, algunos enfoques de AC de *slices* están basados en DRL (Aprendizaje por Refuerzo Profundo), técnica que soluciona los problemas de escalabilidad presentados por el aprendizaje por refuerzo a través del uso de DNN (*Deep Neural Networks*) [30]. Específicamente, [31–33] hacen uso de esta técnica

³MDP hace referencia a la toma de decisiones, basadas en una observación periódica o continua de un sistema dinámico estocástico [29].

para realizar un AC orientado a incrementar el lucro del InP, pero no consideran el estricto cumplimiento de la latencia requerida por una cirugía remota.

Ningún mecanismo de AC propuesto hasta el momento, considera el equilibrio entre la satisfacción de los estrictos requisitos de latencia para una cirugía remota y la optimización del lucro. Razón por la cual este trabajo de grado aborda la siguiente pregunta de investigación:

¿Cómo realizar un control de admisión de *slices* de cirugías remotas en el núcleo de una red 5G para optimizar la latencia y el lucro de los InP?

Para responder esta pregunta de investigación se presentan los siguientes objetivos:

1.2 Objetivos

1.2.1 Objetivo General

Proponer un mecanismo de Control de Admisión de *slices* en el núcleo de red para maximizar la relación latencia/lucro para el servicio de cirugía remota.

1.2.2 Objetivos Específicos

- Diseñar un mecanismo de control de admisión de *slices* basado en aprendizaje por refuerzo profundo para la optimización de latencia y lucro en una cirugía remota.
- Implementar un prototipo del mecanismo propuesto.
- Evaluar mediante simulaciones la eficiencia del prototipo en términos de latencia y lucro.

1.3 Aportes y Envío

1.3.1 Aportes

- Un mecanismo de control de admisión de *slices* que hace uso de técnicas de aprendizaje por refuerzo profundo para la optimización de la latencia y el lucro en el servicio de cirugía remota.
- Un prototipo simulado que implementa el mecanismo propuesto.
- Un análisis de rendimiento en un ambiente simulado del mecanismo propuesto en términos de latencia y lucro.

1.3.2 Envío

El trabajo presentado en esta monografía fue enviado a la comunidad científica a través de un artículo a una revista indexada (Anexo A).

- **Santiago de Jesus Martinez Semanate, Jose Luis Rincón Zapata, Oscar Mauricio Caicedo Rendon and Johanna Andrea Hurtado Sanchez. Admission Control based on Deep Reinforcement Learning for Remote Surgery in 5G.**
 - Estado: Enviada.
 - Clasificación: A2
 - Factor de Impacto: 0.729
 - Índice H: 26 Scimago

1.4 Estructura del Documento

Este documento se divide en los capítulos descritos a continuación.

- Capítulo 1 presenta la **Introducción** que contiene definición del problema, objetivos, Aportes y Envío, y la estructura de este documento.
- Capítulo 2 presenta los **Conceptos Fundamentales** de la investigación. Estos conceptos son esenciales para entender el desarrollo del presente trabajo.
- Capítulo 3 presenta los **Trabajos Relacionados**.
- Capítulo 4 presenta la **Arquitectura del Mecanismo de Control de Admisión basado en DRL**.
- Capítulo 5 presenta la **Evaluación del Mecanismo de Control de Admisión basado en DRL**. Para esta evaluación se define un prototipo, un escenario de prueba y el análisis de los resultados.
- Capítulo 6 presenta las principales **Conclusiones** obtenidas con el desarrollo de este trabajo de grado y **Trabajos Futuros**.

Capítulo 2

CONCEPTOS FUNDAMENTALES

Este capítulo presenta los conceptos que fundamentan la implementación de una cirugía remota como parte de los servicios verticales de las redes 5G.

2.1 Redes de Quinta Generación

Las redes 5G están destinadas a potenciar transformaciones socio-económicas sostenibles de innumerables maneras en la parte productiva y de bienestar [34], presentan mayores capacidades que las generaciones anteriores [35]. 5G asegura altas velocidades de datos, comunicación entre dispositivos cotidianos con diversos requerimientos y conexiones confiables en tiempo real con una mayor eficiencia energética [36, 37]. Además, 5G presenta una arquitectura flexible capaz de soportar múltiples servicios de interacción entre humanos y máquinas con grandes anchos de banda [35].

Las redes 5G comprenden un conjunto diverso de casos de uso para la implementación de nuevos servicios avanzados de telecomunicaciones, los cuales son definidos y clasificados por la ITU-R (*International Telecommunications*

Union—Radiocommunication Sector) de la siguiente manera:

- **eMBB (*enhanced Mobile BroadBand*):** Caso de uso centrado en aplicaciones como transmisión de vídeo de alta definición, realidad aumentada y conexión de eventos a gran escala, caracterizados por el uso de un ancho de banda móvil extenso, con gran velocidades de datos y una mayor cobertura de grandes densidades de usuarios.
- **URLLC:** Este caso de uso 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 (*massive Machine Type Communications*):** Este caso de uso 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 bajo tráfico de datos sin sensibilidad al retraso.

Existen tecnologías habilitadoras que permiten el cumplimiento de los requisitos de 5G, una de ellas es NSL [38], la cual permite el diseño, personalización e implementación de *slices*. NSL soporta múltiples servicios de red con diferentes QoS en una misma infraestructura física, esto produce un mejor uso de los recursos de la red [23–25]. Adicionalmente, tecnologías como SDN (*Software Defined Networking*) y NFV (*Network Function Virtualization*) [38] permiten la implementación de NSL y contribuyen a 5G, habilitando la virtualización de funciones de red y permitiendo un despliegue de aplicaciones y servicios de red con una gran flexibilidad y escalabilidad para diversos servicios.

2.2 Redes Lógicas

La ITU-T (*International Telecommunications Union—Telecommunication Sector*) define NSL como una red lógica que proporciona capacidades y características de

red específicas a segmentos lógicos de red conocidos como *slices*. Los *slices* permiten la creación de redes personalizadas que proporcionan soluciones flexibles para diferentes servicios 5G con diversos requisitos, con respecto a las funcionalidades, el rendimiento y los recursos [12]. NSL permite tener una mejor utilización de los recursos de la red proporcionando solo los recursos necesarios para cumplir los requisitos de QoS de los servicios ofrecidos por un InP [18].

Dada la amplia gama de servicios presentes en 5G, con diversos y estrictos requisitos de QoS, la administración de los recursos representa una tarea desafiante, la cual debe ser diseñada adecuadamente para mejorar el rendimiento de la red [18]. La existencia de una red capaz de configurarse a demanda según el servicio en cuestión, refleja una estructura más flexible y programable orientada al servicio [23]. Esto lleva naturalmente a la necesidad de contemplar el concepto de NSL dentro de las redes 5G, el cual está soportado por tecnologías habilitadoras como SDN y NFV [22, 39].

SDN es una arquitectura emergente de red que logra la adaptabilidad, dinamismo, manejabilidad y rentabilidad de la red [40]. Esto contribuye a la flexibilidad y escalabilidad requerida por NSL para la efectiva satisfacción de los diversos requisitos de 5G. La arquitectura de SDN está constituida por tres capas: plano de control, plano de datos y plano de aplicaciones [41], presentando beneficios como una gestión centralizada, un diseño ágil, una configuración programable y una implementación abierta. De esta manera, los administradores y aplicaciones de red poseen una visión global de los recursos de red, facilitando la planificación y despliegue de *slices* de red.

Una característica esencial de NSL es la capacidad de aprovechar eficientemente los recursos limitados que son brindados por un InP. Este propósito requiere el uso de técnicas robustas de virtualización brindadas por NFV. La ETSI (*European Telecommunications Standards Institute*) define NFV como el principio de separar las funciones de red del *hardware* en el cual son ejecutadas, haciendo uso de la abstracción de *hardware* virtual [42]. En otras palabras, NFV permite migrar las funciones de red (p. ej. firewalls, sistemas de prevención de intrusiones, balanceadores de carga, etc.) basadas en *hardware* a dispositivos de *software*, aprovechando las tecnologías

de virtualización de IT (*Information Technologies*)¹ estándar [44]. El concepto de NFV junto a NSL permite dividir la infraestructura física, usualmente compartida por múltiples usuarios, virtualizando sus recursos físicos y distribuyéndolos adecuadamente según las necesidades específicas de los servicios.

SDN y NFV le proporcionan a NSL una gran flexibilidad en la partición de arquitecturas de red en elementos virtuales. Esto habilita la coexistencia de diferentes servicios verticales sobre la misma infraestructura física [23]. Por lo tanto, NSL habilita la creación de múltiples *slices* sobre una infraestructura física compartida haciendo uso de SDN y NFV. Cada *slice* contempla requisitos de QoS para un servicio particular, como las demandas de latencia presentadas por el servicio de cirugía remota, sin afectar el desempeño de otros *slices* desplegados en la misma infraestructura de red [39].

2.3 Cirugía Remota

Dentro de URLLC se han identificado una variedad de campos de estudio emergentes de las redes 5G [45]. Un caso especial es el sistema de salud, donde la salud tradicional se ha visto revolucionada a través de sistemas inteligentes [46, 47], los cuales promueven servicios novedosos como la cirugía remota [48].

La cirugía remota presenta la capacidad de conectar pacientes y médicos en lugares distantes a través de un sistema de comunicación inalámbrica en tiempo real por medio de 5G [2, 3]. La cirugía remota contribuye a mejorar una serie de problemáticas como la escasez de cirujanos y la falta de atención quirúrgica de alta calidad. Además, evitan la gran cantidad de tiempo, esfuerzo y dinero invertido en viajes de larga distancia. La cirugía remota comprende una arquitectura base compuesta por tres dominios (maestro, red y esclavo), los cuales de manera conjunta permiten realizar el procedimiento quirúrgico [46].

Los dominios presentados para la arquitectura de una cirugía remota dan una visión

¹IT son las tecnologías necesarias para la gestión y transformación de la información, y en particular el uso de ordenadores y programas que permiten crear, modificar, almacenar, proteger y recuperar esa información [43].

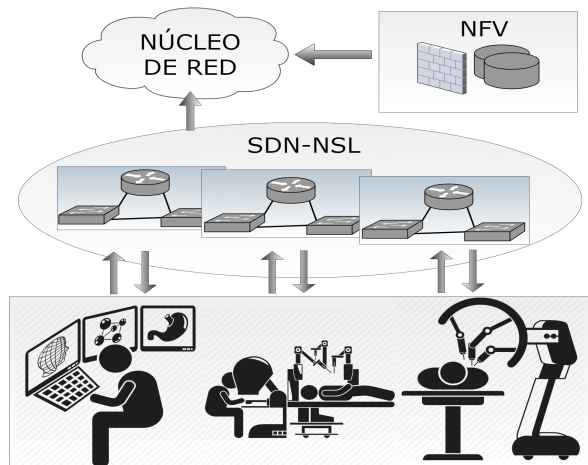


Figura 2.1: Tecnologías habilitadoras de 5G para una cirugía remota

general de la estructura de red requerida para un procedimiento de este tipo. El dominio maestro abarca todos los componentes que facilitan la intervención del cirujano a distancia. El dominio de red representa la infraestructura 5G para el transporte de los datos hasta el lugar del paciente. El dominio esclavo corresponde a la parte sensorial y robótica que recolecta características del entorno y emula los movimientos realizados por el cirujano. En este orden de ideas, se necesita contar con una red capaz de asegurar gran fiabilidad, baja latencia y una correcta adaptación motriz. Este tipo de redes hacen uso de tecnologías habilitadoras como SDN, NFV y 5G NSL, las cuales buscan mejorar el desempeño de los parámetros de QoS [39] (ver figura 4.1). A esto se suma el uso de técnicas que contribuyen en el buen manejo de los recursos en redes destinadas a los servicios *Health-Care*, entre ellas el AC. Esta técnica tiene como objetivo configurar una serie de políticas para el buen manejo de los recursos de una red [49].

2.4 Control de Admisión

AC se define como el conjunto de acciones tomadas durante la fase de establecimiento de un servicio para verificar si una petición debe ser admitida o rechazada [49] (ver figura 2.2). Su principal objetivo es la discriminación de dichas peticiones considerando políticas derivadas de criterios como el lucro o requisitos de QoS como

Casos de Uso 5G	Servicios	Latencia total (ms)	Tasa de Transferencia
URLLC	Cirugía Remota	1	>1 Gbps
	Conducción Autónoma	1~3	>1 Gbps
	Seguridad Pública	5	>1 Gbps
eMBB	**	5~10	>100 Mbps <1Gbps
mMTC	**	>10	1 Mbps

Tabla 2.1: Requisitos latencia y ancho de banda de los casos de uso 5G

la latencia [50]. De esta manera, un AC enfocado en el servicio de cirugía remota únicamente admite peticiones que logran contribuir positivamente en el cumplimiento de latencia de este servicio.

Las políticas que gobiernan un mecanismo de AC no se rigen por un estándar, ellas se encuentran directamente relacionadas con las necesidades del InP y los recursos disponibles para satisfacer los requerimientos de QoS de los usuarios (p. ej. proveedores de servicios *Cloud* o telemedicina) que hacen parte de la red [51]. La implementación de políticas efectivas sobre un mecanismo de AC genera grandes beneficios, potenciando el uso eficiente de los recursos de la red [49].

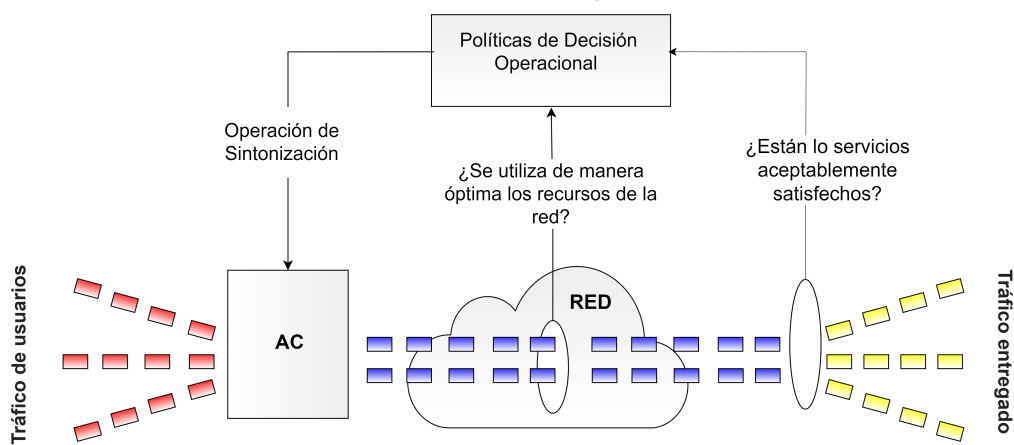


Figura 2.2: Ejemplo de control de admisión

Hace poco menos de dos décadas, algunas investigaciones aseguraban que ningún esquema de AC era capaz de proporcionar garantías estrictas de QoS, pues las segmentaciones personalizadas y completas de los recursos de red en el núcleo eran complejas e increíblemente costosas [52]. Sin embargo, conceptos como SDN, NFV y NSL, fomentan la creación de mecanismos de AC de *slices* capaces de solventar los requisitos QoS presentes en los casos de uso de 5G [3]. 5G y sus tecnologías habilitadoras, permiten la creación de esquemas de AC de *slices* inteligentes, flexibles y adaptables, capaces de satisfacer las estrictas exigencias presentadas por los servicios de la industria vertical²[3], en especial servicios como la cirugía remota, los cuales presentan una notable sensibilidad al retraso debido a su interacción en tiempo real [54]. Los mecanismos de AC de *slices*, soportados por el concepto de NSL, son fundamentales en el despliegue de servicios 5G y claves de este trabajo de grado.

2.5 Aprendizaje por Refuerzo Profundo

ML (*Machine Learning*) presenta todo un ecosistema de técnicas con el objetivo principal de aprender de los datos, identificar patrones ocultos y tomar decisiones con poca o mínima intervención humana [55]. DRL, como parte de dicho ecosistema, representa un paso hacia la construcción de sistemas autónomos con una comprensión de mayor nivel del entorno [30], logrando un impacto significativo en el área de las comunicaciones y redes [26]. No obstante, para comprender su funcionamiento es necesario abarcar técnicas tanto de RL como de DL (*Deep Learning*).

RL difiere significativamente de otros métodos de ML, construyendo y actualizando un modelo a partir de la interacción con su propio entorno [56]. RL es un proceso de aprendizaje donde un agente puede tomar decisiones periódicamente, observar los resultados y luego ajustar automáticamente su estrategia para alcanzar una política óptima [26]. Por ejemplo, un agente RL puede construir una política óptima de aceptación de clientes que le permita optimizar el lucro obtenido por un InP. RL

²La industria vertical abarca todos los sectores de negocio que utilizan o utilizarán infraestructura de red para la entrega de sus servicios y/o cumplimiento de sus funciones [53].

posee características distintivas como la búsqueda de ensayo/error y la recompensa [57]. La primera, describe una fase de entrenamiento basada en la experiencia, donde se recibe una retroalimentación del entorno para aprender la secuencia correcta de acciones donde la recompensa es maximizada [58] y la política optimizada. Por ejemplo, si un agente busca optimizar el lucro en una tarea de aceptación de clientes, la recompensa será el lucro obtenido por cada cliente aceptado y la política optimizada será el conjunto adecuado de clientes que le brinden la máxima recompensa, es decir, el máximo lucro. La segunda, sugiere que RL puede predecir e influir en situaciones futuras, ajustándose adecuadamente a problemas con soluciones a largo plazo. Este modelo de ML puede implementarse a través de algoritmos de Monte Carlo, Dyna-Q, Fuerza Bruta o Q-learning [57], siendo este último el más utilizado. La naturaleza de RL ha permitido una contribución enorme en la solución de problemas de naturaleza aleatoria y no estacionaria, tal es el caso de las redes de computadores, donde se presentan aportes en enrutamiento [59], programación de recursos en centros de datos [60] y seguridad informática [61].

DL como RL, representa una pieza fundamental en el desarrollo del DRL. DL es una técnica del ML que modela abstracciones de alto nivel en los datos mediante el uso de ANN (*Artificial Neural Networks*) [62]. Su modelo de aprendizaje está compuesto de múltiples capas consecutivas de procesamiento que permiten la extracción jerárquica de características [63]. Toda su estructura está inspirada en un cerebro biológico [64], donde gran cantidad de unidades de procesamiento de información (llamadas neuronas) permiten manejar un extenso flujo de información. Las características presentadas por DL son adecuadas para la solución de desafíos relacionadas con el creciente consumo de tráfico IP de las redes actuales, la gestión de recursos [65] y los sistemas inteligentes de enrutamiento [66]. Además, DL es ampliamente usado en áreas de transporte inteligente para control de tráfico [67, 68] y modelos de reconocimiento facial para detección de rostros [69].

RL y DL fueron articulados para construir la técnica denominada DRL. DRL cumple la filosofía de un agente que interactúa con el entorno y aprende una política óptima a través de ensayo y error [57]. Sin embargo, en contraste con RL, donde toda la información es almacenada haciendo uso de una representación tabular [57], DRL adopta el uso de una DNN para optimizar el proceso de aprendizaje con DL, mejo-

rando así la velocidad y el rendimiento de los algoritmos de aprendizaje [26]. Como resultado, surge el algoritmo DQL (*Deep Q-Learning*) [70], del cual derivan técnicas con modelos avanzados como: DDQL (*Double DQL*) [71], *DQL With Prioritized Experience Replay* [72], *Dueling DQL* [73], *Asynchronous Multi-Step DQL* [74], *Distributional DQL* [75], *DQL With Noisy Nets* [76] y *Rainbow* [77]. Además, técnicas de DRL han implementado extensiones de MDP como: *Deep Deterministic Policy Gradient Q-Learning for Continuous Action* [78], *Deep Recurrent Q-Learning for POMDPs* [79], *Deep SARSA Learning* [80] y *DQL for Markov Games* [81]. Los enfoques de estas técnicas presentan soluciones más sofisticadas, manifestando mejoras significativas en la velocidad de aprendizaje con grandes espacios de estados y la construcción de redes más seguras y solidas [26]. Estas características han permitido su implementación en diversos campos incluyendo enrutamiento en redes informáticas [82–84], sistemas de navegación para robots autónomos [85, 86] y en el área de los juegos de vídeo [87–89]. Recientes investigaciones revelan que el uso de DRL contribuye significativamente a los mecanismos de AC dentro del marco de NSL [32, 33, 90], esto permite una asignación de *slices* eficiente, inteligente y rentable, logrando un mejor uso de los recursos y un menor tiempo de convergencia de los algoritmos [26].

Capítulo 3

TRABAJOS RELACIONADOS

Este capítulo presenta la investigación sobre el control de admisión de *slices*. La Tabla 3.1 resume brevemente la descripción de las propuestas que se encuentran en la literatura y sus características. A continuación, señalamos brevemente sus principales aportes y diferencias con respecto a nuestro trabajo de grado.

3.1 Control de Admisión con Algoritmos Heurísticos

Los autores de [18] presentan un AC eficiente que realiza un método de priorización heurística. Proponen dos algoritmos 5G-SA y 5G-AC-SA para la toma de decisiones con base en los requisitos de QoS de servicios: *TV Streaming*, *Car Management* y *Web Browsing*. Los algoritmos aceptan una petición de *slices* cuando los recursos disponibles son suficientes para garantizar la tasa de transferencia de datos mínima en cada servicio. Los autores concluyen que la solución genera un aumento de la QoE (*Quality of Experience*) con una mejor gestión de los recursos de red en comparación con el algoritmo 4G-SA. Sin embargo, la solución presentada está limitada a la aceptación de un solo *tenant* por acción, lo que podría generar congestión en una situación *multi-tenant* donde se necesita atender gran cantidad de peticiones para *slices* de red.

La solución presentada en [19], está basada en *Overload Cost and Requirements*, un algoritmo heurístico concebido para reducir los costos de implementación de *slices*, sin afectar la fiabilidad, disponibilidad y latencia del servicio. El mecanismo encuentra entre los *slices* existentes, aquellos que cumplan las características de QoS del servicio y haciendo uso de una función de costo, calcula la tasa de sobrecarga de cada recurso de red en el *slice* y selecciona la más baja. Los resultados de la evaluación muestran una clara disminución en el costo de implementación, manteniendo estables los parámetros de fiabilidad, disponibilidad y latencia. Sin embargo, el uso de heurísticas podría limitar la capacidad del mecanismo en términos de adaptabilidad y toma de decisiones, dada su reducida habilidad para detectar patrones y analizar profundamente los estados de una red.

Los autores en [20] presentan un AC para la asignación de recursos en RAN y núcleo de red, para maximizar el lucro de un MNO (*Mobile Network Operator*). El problema de optimización introducido en esta solución posee la forma de programación booleana lineal, ideal para *Knapsack Problem*¹. Los autores presentan un algoritmo subóptimo de dos pasos. En cada paso del algoritmo propuesto, resuelven una heurística para una variante del *Knapsack Problem*. Al final los autores evidencian un AC con una capacidad de manejo y asignación de los recursos de manera simultánea en el núcleo y la RAN, logrando obtener un mejor desempeño que otras propuestas basadas en algoritmos heurísticos en términos de lucro. A pesar de ser una solución con alto desempeño, las desventajas se dan en las políticas de aceptación de las peticiones de *slices*, las cuales no tienen en cuenta parámetros críticos de 5G como latencia.

Los autores de [21] consideran colas heterogéneas con diferentes prioridades diseñando un controlador para un AC, el cual permite la máxima utilización de recursos de red basados en dos enfoques. El primero considera, el orden de llegada. El segundo, el tipo de *slice*. El controlador realiza una clasificación de *tenants* pacientes (prioridades bajas) e impacientes (prioridades altas) por medio de modelado estadístico. Los *tenants* impacientes tendrán una prioridad más alta sobre los *tenants* pacientes. Aunque sus resultados son sobresalientes, el AC propuesto no

¹El *Knapsack Problem* es un problema clásico de optimización combinatoria, que buscan la mejor solución entre varias soluciones disponibles [91].

considera los diversos casos de uso de 5G, ignorando la variedad de QoS presentada por sus servicios emergentes.

En general, las propuestas heurísticas presentan beneficios en términos de disminución de costos de implementación, utilización y asignación de recursos. Sin embargo, suelen tener limitaciones predictivas, lo que reduce su capacidad de análisis en la toma de decisiones. Por otro lado, las propuestas presentan políticas de aceptación que afectan su desempeño en términos de satisfacción de latencia y lucro, dado que se acepta un *slice* por acción o según su orden de llegada, esto impide la priorización de servicios de alta demanda de QoS como los requeridos por el servicio de cirugía remota.

3.2 Control de Admisión Basado en Aprendizaje por Refuerzo

En [27] se presenta una solución para la admisión y asignación de múltiples peticiones de *slices* para servicios de 5G. Maximizando el lucro para un InP y asegurando la QoS para los *tenants*. Esta solución es modelada por medio de un SMDP (*Semi-Markov Decision Process*)² y el uso de un algoritmo adaptativo de *Q-Learning*, el cual permite la creación de políticas para la aceptación y asignación de *slices* de red. La solución presenta un rendimiento óptimo en términos de la relación entre la QoS y el lucro del InP, exponiendo resultados superiores a soluciones heurísticas similares, no obstante, el enfoque implementa el algoritmo *Q-Learning*, el cual posee una escalabilidad limitada debido a que las actualizaciones de la tabla Q ³ se vuelven complejas con el incremento del espacio de estados y acciones. Además, los requisitos de QoS de una cirugía remota no se tienen en cuenta para el diseño de este mecanismo.

La propuesta [13] presenta un AC para permitir que los MNO aprovechen al máximo sus recursos. Haciendo uso de modelos de predicción HW (*Holt-Winters*), un agente

²SMDP es un tipo especial de MDP apropiado para modelar sistemas de eventos discretos de tiempo continuo [92].

³Representación tabular que almacena los valores Q de cada par estado-acción.

RL predice el comportamiento del tráfico generado por 5 tipos de *tenants* que difieren en sus características de QoS, analizando patrones de movilidad en las celdas basados en el modelo SLAW (*Self-similar Least Action Walk*). La información de predicción de tráfico y los patrones de movilidad, junto a los valores de lucro de cada petición, conforman los parámetros que el proceso de AC (haciendo uso de RL) tendrá en cuenta para fijar una lista de los *tenants* que podrán ingresar a la red. Los resultados obtenidos por esta propuesta son significativamente buenos, logrando potenciar la utilización de los recursos, no obstante, todos sus esfuerzos se enfocan en la RAN y no se contempla el servicio de cirugía remota. De esta manera, el enfoque presentado difiere de la idea central de este trabajo de grado, la cual busca soluciones en el núcleo de la red enfocadas específicamente al servicio de cirugía remota.

Los autores de [28] proponen y evalúan un modelo de estados para un AC síncrono⁴ de *slices* para un MNO. El mecanismo propuesto se basa en un MDP asíncrono, donde cada estado contempla las diferentes peticiones de asignación de recursos y se establece una región de admisibilidad, la cual representa el conjunto de peticiones que cumplen con una estrategia de admisión establecida. La estrategia asegura la aceptación de peticiones únicamente si los recursos exigidos no superan los recursos disponibles. Para agilizar este proceso, los autores utilizan un modelo analítico para predecir los valores de transición del MDP, lo cual reduce la complejidad computacional presentada por el mecanismo al contemplar múltiples *slices*. No obstante, debido a la implementación de una estrategia estática, su complejidad varía e incrementa junto con el aumento del número de *slices*, de manera que se presenta como una solución inviable debido a su escalabilidad limitada.

Las propuestas basadas en RL presentan políticas de decisión inteligentes en la aceptación de *slices*, la maximización del lucro, la satisfacción de QoS, la utilización de los recursos y la complejidad computacional. No obstante, entre sus deficiencias se puede mencionar la limitada escalabilidad y la ausencia de priorización de servicios con estrictos requisitos de latencia, como la cirugía remota.

⁴AC síncrono toma decisiones de ingreso periódicas en lugar de inmediatamente cuando llegan las peticiones [28].

3.3 Control de Admisión Basado en Aprendizaje por Refuerzo Profundo

Los autores de [31] presentan un AC en la RAN basado en la clasificación de dos tipos de *slices*. El primer tipo considera todos los *slices* con alta prioridad, los cuales manejan tiempos de latencia estrictos y alto lucro. El segundo tipo contempla todos los *slices* con baja prioridad que manejan tiempos de latencia no estrictos y bajo lucro. El AC está basado en un agente RL que acepta o rechaza las peticiones de *slices* mediante el análisis de una ANN, la cual recibe una matriz con la cantidad de recursos disponibles. Esta propuesta presenta mejoras en comparación a soluciones heurísticas similares, demostrando que las políticas usadas poseen la capacidad de adaptarse a diferentes condiciones. Los autores obtienen resultados con un mayor rechazo de peticiones de *slices* de bajo lucro, debido a la búsqueda de un mayor lucro. Sin embargo, su decisión de admisión se realiza en orden de llegada, de manera que solo se tiene en cuenta una petición por acción, lo que dificulta el manejo de grandes grupos de peticiones de *slices*.

Los autores de [32] desarrollan un algoritmo adaptativo *Network-slicing Neural Network Admission Control* basado en una red neuronal de DRL. La solución está diseñada para lograr un balance entre QoS y lucro del InP. El algoritmo presenta varias ventajas comparado con algoritmos de *Q-Learning*, los cuales llegan a requerir una mayor cantidad de iteraciones para su convergencia. Esto permite un aprendizaje del entorno más rápido y efectivo que otra clase de algoritmos y por tanto, pueda ser usado en redes con gran flujo de información. El algoritmo propuesto brinda gran escalabilidad dada la cantidad de *slices* que pueden ser soportados al tiempo, convirtiéndose así en una solución óptima para el InP. Sin embargo, el algoritmo no considera como parámetros los estrictos requisitos de latencia de una cirugía remota para creación y asignación de los *slices*.

La solución presentada en [33] contempla el aprovisionamiento de recursos de *slices* y su AC. Su mecanismo evalúa tres tipos de *tenant* con KPIs(*key Performance Indicator*) que aseguran el buen rendimiento del servicio. El esquema propuesto está basado en la implementación híbrida de una DQN (*Deep Q-Network*) y una

matriz jacobiana⁵, las cuales permiten calibrar parámetros de control como el límite de recursos que un *slice* puede ofrecer. El proceso de AC permite la entrada de los usuarios mientras los recursos necesarios no superen el límite de recursos ofrecidos por el *slice*. Las recompensas de la DQN están basadas en el cumplimiento o la violación de los KPI, esto permite una rápida convergencia hacia la política más óptima. No obstante, aunque encuentra políticas óptimas enfocadas en los KPI, no considera el lucro del InP.

Las propuestas basadas en DRL expanden su capacidad computacional y predicativa en contraste con RL, debido al uso de DNN. Esto les permite optimizar las estrategias de aceptación, con el fin de maximizar el lucro y el cumplimiento de los requisitos de QoS. Sin embargo, entre las desventajas detectadas se puede mencionar la aceptación de un solo *slice* por tiempo y no contemplar los estrictos requerimientos de QoS demandados por el servicio de cirugía remota.

3.4 Brechas

Ref.	Técnica de AC	Casos de Uso 5G	Red		Propósito
			Núcleo	Acceso	
[18]	Heurística	✓		✓	Satisfacer QoE en 5G
[19]		✓	✓	✓	Maximizar el lucro de InP, Garantizar fiabilidad, disponibilidad y latencia del servicio
[20]			✓	✓	Mejorar la gestión de recursos de un MNO
[21]			✓	✓	Maximizar el lucro del InP
[27]	RL		✓		Maximizar el lucro del InP
[13]				✓	Maximizar el lucro del InP y Garantizar los requisitos de QoS
[28]			✓		Mejorar la gestión de recursos de un MNO
[31]	DRL		✓		Maximizar el lucro del InP
[32]		✓	✓		Maximizar el lucro del InP
[33]				✓	Garantizar KPI's de cada usuario
Pro.	DRL	✓	✓		Optimizar el lucro bajo los estrictos requisitos de latencia de el servicio de Cirugía Remota

Tabla 3.1: Trabajos relacionados

Algunos mecanismos de AC para *slices* han propuesto soluciones basadas en heurísticas [18–20] y la teoría de colas [21], las cuales carecen de la flexibilidad suficiente para adaptarse a la variabilidad de las redes 5G. Otras soluciones utilizan

⁵Derivada del vector de error de salida con respecto a parámetros ajustables (no lineales) [93].

técnicas de BDA (*Big Data Analytics*) [94] y RL [13, 27, 31], ofreciendo flexibilidad e inteligencia en la toma de decisiones, pero son limitadas de forma predictiva y computacional. Las soluciones basadas en DRL [32, 33] solventan las desventajas mencionadas previamente, sin embargo su enfoque difiere de este trabajo de grado dado que no existe una priorización con respecto al servicio de cirugía remota. Todos estos mecanismos abarcan problemas relacionados con la tasa de admisión, la utilidad de la red, la QoE, la QoS, la disponibilidad, las ganancias monetarias, el *throughput* y la confiabilidad. Sin embargo, aunque algunos se centran en el cumplimiento de la latencia, ninguno contempla los requisitos de latencia altamente estrictos y específicos del servicio de cirugía remota y su equilibrio con la optimización de lucro de los InP.

El mecanismo de AC propuesto en este trabajo de grado aprovecha las propiedades de DRL para mitigar las deficiencias en términos de flexibilidad, capacidad predictiva y capacidad computacional que presentan las propuestas anteriormente mencionadas. Este mecanismo de AC basado en DRL desarrolla políticas inteligentes de admisión de *slices* que permiten un equilibrio adecuado entre el cumplimiento de los requisitos de latencia del servicio de cirugía remota y la optimización de la ganancia de InP. Sin olvidar que el entorno considera los diversos casos de uso necesarios para un ecosistema 5G.

Capítulo 4

Control de Admisión basado en Aprendizaje por Refuerzo Profundo para Cirugía Remota en 5G

Este capítulo presenta los elementos que componen el mecanismo AC propuesto. La sección 4.1 presenta la arquitectura del mecanismo de AC y sus respectivos módulos. La sección 4.2 presenta las características internas del agente DRL. Finalmente, la sección 4.3 presenta el algoritmo del mecanismo de AC propuesto.

4.1 Arquitectura

4.1.1 Motivación

En el ecosistema de NSL, un InP recibe múltiples peticiones proveniente de múltiples *tenant* para la creación de *slices* en un período de tiempo. Estas peticiones incluyen una amplia variedad de requisitos de QoS de los diversos casos de uso de 5G. La

satisfacción de estos requisitos de QoS se puede abordar a través de un mecanismo de AC que sigue las pautas de cada InP específico. Por esta razón, es necesario crear una arquitectura del mecanismo AC que permita implementar las políticas de un InP específico.

Cumplir con los estrictos valores de latencia de los *slices* para cirugía remota es uno de los desafíos más importantes para un 5G InP. La latencia de una Cirugía Remota está directamente relacionada con el dinamismo del entorno, el cual está constituido por todos los movimientos que realiza el cirujano y las re-alimentaciones hápticas que logra percibir durante el procedimiento quirúrgico [45]. Por tanto, se establecen dos tipos de entornos: entornos altamente dinámicos y entornos medianamente dinámicos. Los entornos altamente dinámicos hacen uso de dispositivos táctiles, que permiten percibir la textura, temperatura, presión o vibración del cuerpo [95]. Mientras que los entornos medianamente dinámicos utilizan dispositivos cinestésicos para la percepción del peso, las fuerzas ejercidas sobre el cuerpo y la ubicación espacial [95]. La Cirugía Remota que pertenece a un entorno Altamente dinámico se denomina RS (Cirugía remota Háptica), mientras una Cirugía Remota Convencional pertenece a un entorno medianamente dinámico.

Estructurar una arquitectura para un mecanismo AC con políticas que prioricen servicio de cirugía remota es fundamental para satisfacer las necesidades de latencia que demanda este servicio. Las arquitecturas para un mecanismo AC en 5G deben considerar los diversos casos de uso, pero aquellas enfocadas en el servicio cirugía remota deben promover de manera especial el cumplimiento de sus estrictos requisitos de latencia, dado que su incumplimiento puede causar problemas en los procedimientos quirúrgicos. Sin olvidar que las ganancias también son un objetivo común y primordial de los InP, la arquitectura propuesta en este trabajo de grado tiene como objetivo encontrar el equilibrio entre satisfacer los requisitos de latencia de los *slices* para cirugía remota y la optimización del lucro de los InP.

4.1.2 Visión General

La arquitectura del mecanismo de AC propuesto necesita realizar un reconocimiento previo de la infraestructura 5G, para recopilar información de todos los posibles *slices* que podrían crearse para desplegar las SLR (*Slices Request*). Cabe destacar que este proceso es un paso previo al ciclo de trabajo de la arquitectura del mecanismo AC y se realiza una única vez para reconocer la topología de la red. El ciclo de trabajo de la arquitectura comienza cuando se extrae toda la información sobre el estado actual de los elementos que componen la infraestructura 5G (porcentajes de disponibilidad de ancho de banda, CPU y valores de latencia). Posteriormente, múltiples *tenants* envían un conjunto de SLRs y el agente de DRL determinará si serán aceptadas o denegadas en base a la información recopilada previamente. Dado que el conjunto de peticiones pre-admitidas está compuesto por SLR de los diversos casos de uso de 5G, se organizan y priorizan teniendo en cuenta aquellas que generen mayor lucro para un InP como son los servicios de una cirugía remota. A continuación, se lleva a cabo un proceso de validación que verifica si efectivamente la infraestructura 5G cuenta con los recursos necesarios para cumplir con las SLR pre-admitidas. Una vez validadas las SLR, se reservan sus recursos de red, se asigna su vida útil y luego se despliegan en la infraestructura 5G. Posteriormente, se analiza el rendimiento de las SLR desplegadas, con el fin de recopilar información que proporcione retroalimentación al agente DRL y facilite su entrenamiento. Finalmente, el ciclo de trabajo de la arquitectura inicia nuevamente.

4.1.3 Módulos

En esta subsección, presentamos los módulos de la arquitectura del mecanismo AC para *slices* de red, el cual contempla los diversos casos de uso de 5G principalmente el servicio de Cirugía Remota. La Figura 4.1 muestra una arquitectura formada por la Infraestructura de red 5G, el Módulo de Control de Admisión, el Módulo de Validación, el Módulo de Instanciación, el Módulo de Reconocimiento, el Módulo de Monitoreo, y el Repositorio.

La Infraestructura de red 5G se basa en un conjunto interconectado de NFVI-

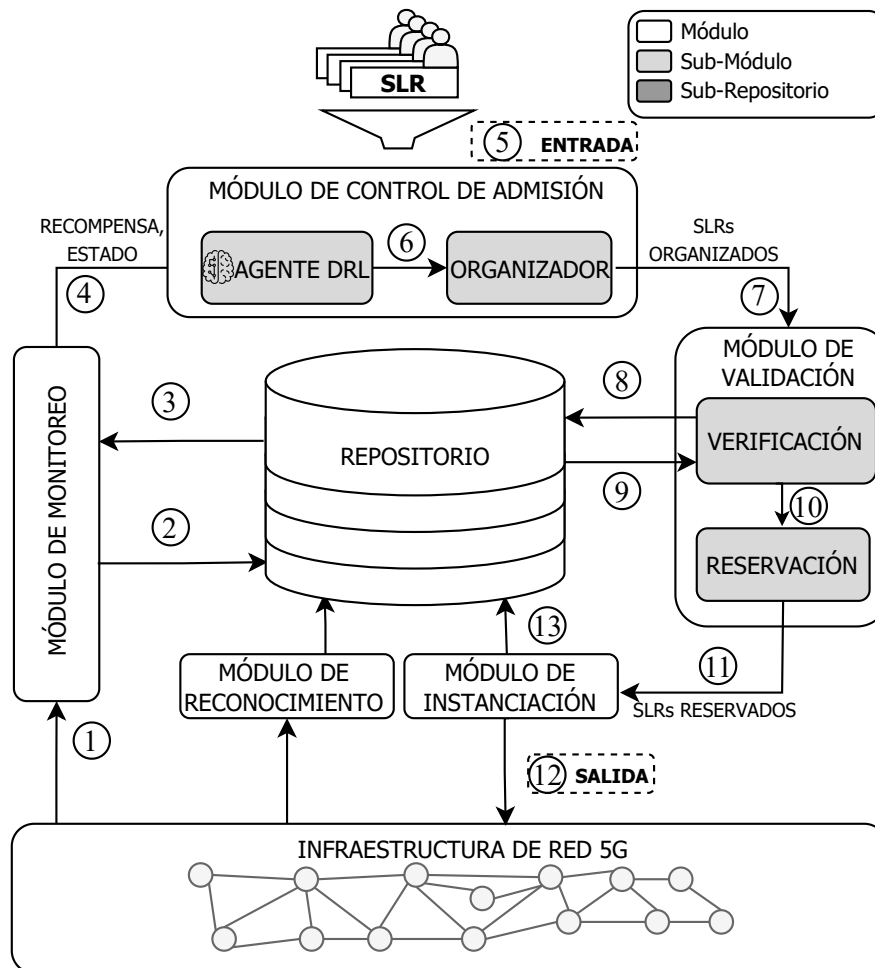


Figura 4.1: Arquitectura del Mecanismo de Control de Admisión

PoPs (*Network Function Virtualization Infrastructure Points of Presence*) [96]. Los NFVI-PoPs son nodos que ofrecen capacidad de procesamiento para el despliegue de múltiples VNF (*Virtualized Network Function*) [97]. Algunos nodos están centralizados (nodos de alta capacidad o nodos centrales) y algunos están descentralizados, cerca del usuario final (nodos de baja capacidad o nodos de borde).

Las SLR, provenientes de múltiples tenants, a implementar en la Infraestructura de Red 5G están compuestas por VNF que deben alojarse en los NFVI-POPs (ver figura 4.2). Para mejorar la escalabilidad y la flexibilidad en la implementación de las VNF,

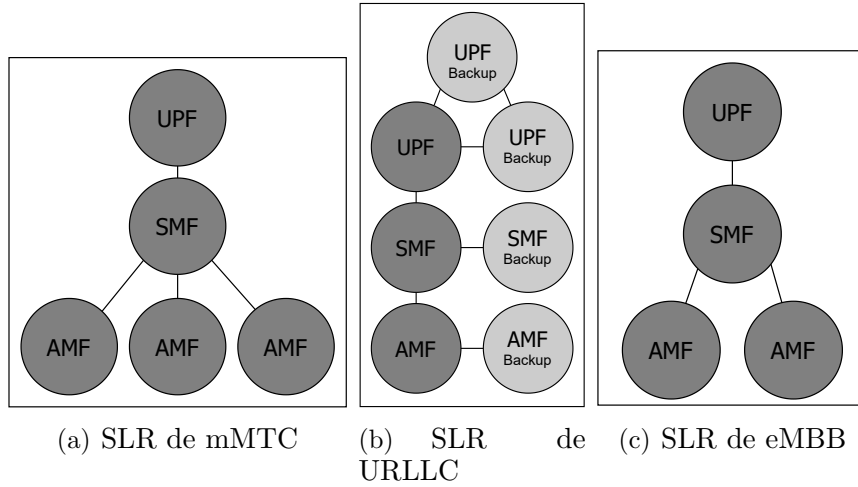


Figura 4.2: Grafos de las SLR

las SLR se diseñaron teniendo en cuenta el concepto de 5G CUPS (*Control and User Plane Separation*) permitiendo implementaciones en ubicaciones centralizadas y de borde[98]. 5G CUPS define la separación entre el CP (*Control Plane*) y el UP (*User Plane*) en el núcleo de la red. De esta manera 5G CUPS, permite diseñar SLR altamente personalizadas (ver figura 4.2), dado que es posible desplegar conjuntos particulares de VNF, pertenecientes al CP o UP, según la demanda del tenant. CP define funciones de red como la AMF (*Access and Mobility Management Function*) y la SMF (*Session Management Function*), mientras que UP define la UPF (*User Plane Function*). Esta propuesta comprende cuatro tipos de SLR: RS, URLLC, eMBB y mMTC, donde cada tipo de SLR tiene un requisito de QoS específico en términos de procesamiento (CPU), ancho de banda (BW) y latencia (p. ej. {Latencia : 0,1 ms, ancho de banda: 4 unidades, CPU: 8 unidades}).

El Módulo de Reconocimiento lleva a cabo la primera tarea de la arquitectura del mecanismo AC. Esta tarea se considera una etapa previa de descubrimiento de red, en donde se analiza la Infraestructura de Red 5G y recolecta la información de todas las posibles combinaciones de nodos y enlaces de la topología que conjuntamente logran cumplir con el diseño topológico requerido por los diferentes tipos de SLR (ver figura 4.2). De esta manera, el Módulo de Reconocimiento puede identificar todos los posibles *slíces* de cada tipo de SLR que pueden ser creados dentro de la Infraestructura de red 5G. Este proceso se lleva a cabo una única vez

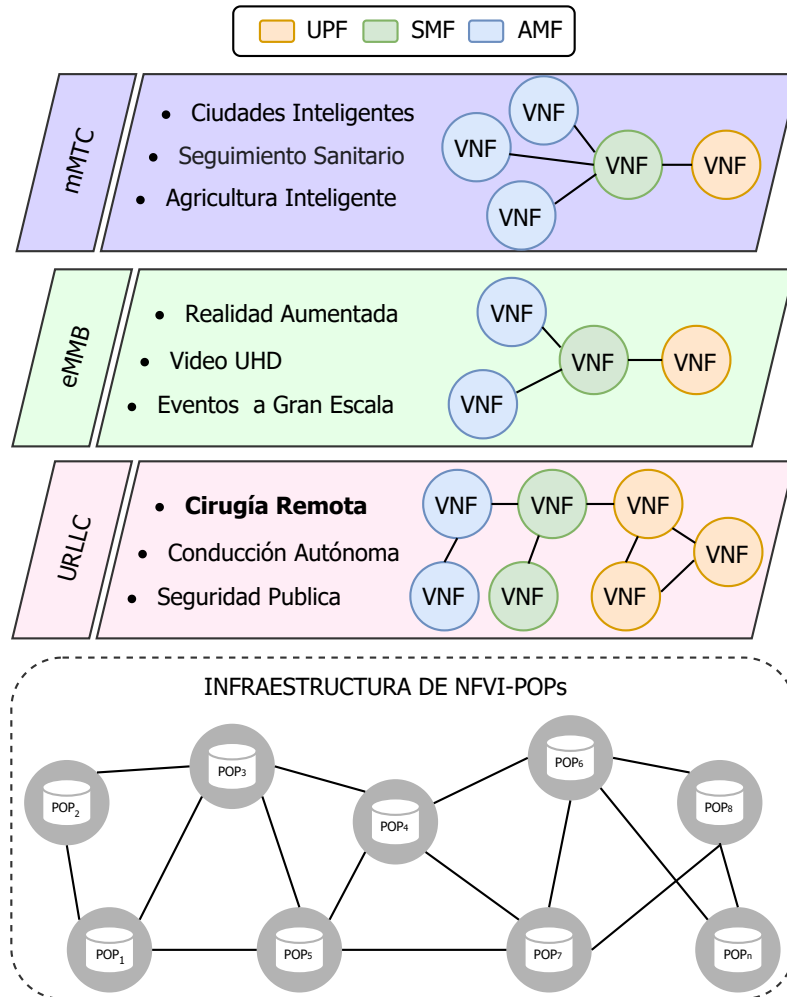


Figura 4.3: Slices para cirugía remota y otras aplicaciones

en la topología de red y almacena la información topológica de cada *slice* en el **Repositorio**, específicamente en el grupo *Slices* Totales, con la siguiente estructura: $Slice_n = \{POP_1 : AMF, POP_{13} : AMF, POP_8 : SMF, POP_4 : UPF\}$. En esta estructura, cada VNF es asignada a un nodo POP, de manera que el conjunto de POPs seleccionados y los enlaces que los unen, cumplan con el grafo requerido por cada tipo de SLR (Ver figura 4.3).

El Módulo de Monitoreo da inicio al ciclo de trabajo de la arquitectura del mecanismo AC, mediante una tarea de recolección de información relacionada con

el estado actual de los elementos que componen la Infraestructura de Red 5G. La información anterior incluye porcentajes de disponibilidad en términos de ancho de banda y CPU de los enlaces virtuales y puntos de presencia ①, respectivamente. El grupo *Slices* Totales recibe esta información ② y determina el conjunto de *slices*, identificados en la etapa de reconocimiento, que cumplen con el ancho de banda, CPU y latencia que son exigidos por cada tipo de SLR. Este conjunto de *slices* se almacena en el grupo *Slices* Usables (el cual hace parte del Repositorio), donde su información es actualizada en cada repetición del ciclo de trabajo dado que el ancho de banda, CPU y latencia en la Infraestructura de Red 5G varían continuamente. La información de cada *Slice* almacenado en el grupo *Slices* Usables, se estructura de la siguiente manera: $Slice_n = (\{POP_1 : AMF, POP_{13} : AMF, POP_8 : SMF, POP_4 : UPF\}, Latencia, Tipo)$. Esta información contiene la asignación de cada VNF a su respectivo nodo POP y adicionalmente, su latencia y el tipo de SLR que podría soportar. Seguidamente, la información del grupo *Slices* Usables se agrega a la información extraída por el Módulo de Monitoreo ③ para calcular el estado actual de la Infraestructura de Red 5G y notificarlo al Módulo de Control de Admisión ④.

El Módulo de Control de Admisión recibe la información del estado actual de la Infraestructura de red 5G y activa una ventana de tiempo para permitir la entrada de un conjunto de SLR ⑤, que representan los elementos de entrada de la arquitectura propuesta para el mecanismo AC. Estas SLR llegan mediante un proceso de *poisson* [99], el cual permite modelar situaciones en las que se necesita determinar el número de hechos de cierto tipo que se producen en un intervalo de tiempo, bajo condiciones de aleatoriedad (ver ecuacion 4.1). El Módulo de Control de Admisión pre-admite un conjunto de SLR mediante dos Sub-Módulos: Agente DRL y Organizador. El agente DRL toma una decisión con base al estado actual, el cual está representado por una tupla de pesos que indican el porcentaje de aceptación de cada tipo de SLR que son pre-admitidas en la red. Por ejemplo, para la tupla $\{0.6, 0.8, 0.3, 0.5\}$ se aceptan 60%, 80%, 30% y 50% de SLR de RS, URLLC, eMBB y mMTC, respectivamente. El agente DRL selecciona la combinación de pesos que conduce a una política óptima en términos de satisfacción de latencia y lucro (el agente DRL se detalla en la sección 4.2), y luego se notifica al Organizador ⑥. El Organizador selecciona y organiza el grupo de peticiones pre-admitidas teniendo en

cuenta la combinación seleccionada por el agente DRL. El organizador estructura una lista denominada SLR organizadas, que son organizadas según dos criterios: tipo de SLR y duración. Debido a que RS y URLLC tienen requisitos de latencia más altos, el Organizador los clasifica en la parte superior del servicio prioritario. Por otro lado, entre el conjunto de peticiones del mismo tipo, se priorizarán aquellas con mayor vida útil, ya que la vida útil de la SLR es directamente proporcional al lucro.

$$f(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (4.1)$$

Donde:

k : Numero de ocurrencias

λ : Media de la distribución

El Módulo de Validación recibe el conjunto de SLR (7) contenidas en la lista SLR organizadas. Este módulo verifica la existencia de recursos de red suficientes para las SLR organizadas y los reserva usando los submódulos verificación y reservación, respectivamente. El Submódulo de verificación analiza los requisitos de latencia presentados por cada SLR y verifica la existencia de cualquier *slice* en el grupo *Slices Usables* que pueda satisfacerlos (8) (9). El Submódulo de reservación recibe el conjunto de SLR del Submódulo de verificación (10) y reserva los recursos de red requeridos. Posteriormente, la lista final de SLR admitidas se envían al Módulo de instanciación (11).

El Módulo de Instanciación tiene tres objetivos: el primero es asignar el tiempo de vida a los recursos según lo solicitado por cada SLR (ver sección 5.1.1) y además, usar dichos recursos para desplegar las VNF de las SLR en la Infraestructura de Red 5G (12). El segundo es almacenar la información de dichas SLR en el grupo de *Slices Desplegados* (13), el cual tiene la siguiente estructura: $Slice_n = (\{POP_1 : AMF, POP_{13} : AMF, POP_8 : SMF, POP_4 : UPF\}, SLR, Tiempo\ de\ vida)$. El tercero es comprobar frecuentemente el tiempo de vida restante de los recursos asignados a las SLR en el grupo *Slices Desplegados*. Cuando alguna SLR del grupo *Slices*

Desplegados completa su tiempo de vida en la red, el Módulo de Instanciación libera dichos recursos. En este punto, la disponibilidad de los recursos en la infraestructura 5G ha cambiado y por consiguiente, el Módulo de Monitoreo calcula los potenciales de satisfacción nuevamente (ver sección 4.2.1).

Finalmente, el Módulo de Monitoreo observa la latencia y el lucro experimentados por las SLR desplegadas ①. Esta información es utilizada para estructurar las recompensas (ver sección 4.2), que se entregan al agente DRL para su respectivo entrenamiento ④. A partir de este momento, el ciclo de trabajo de la arquitectura del mecanismo de AC inicia nuevamente.

4.2 Agente Basado en Aprendizaje por Refuerzo Profundo para el Control de Admisión del Servicio de Cirugía Remota en 5G

Un agente DRL construye una política de decisión óptima que maximiza las recompensas en todo momento y, en consecuencia, conduce a la satisfacción de un objetivo específico. Por lo tanto, un agente DRL puede maximizar el rendimiento de la red para determinadas situaciones, ya que sus criterios de decisión se actualizan en función de la suma ponderada de las recompensas de decisiones en estados anteriores y la expectativa de estados futuros. Esto indica que la recopilación continua de datos de red (proporcionada por el Módulo de Monitoreo) y la toma de decisiones frecuente (ejecutada en el Módulo de Control de Admisión) convergerán en una política que satisface los valores de latencia de los *slices* para cirugía remota y optimiza el lucro del InP.

A diferencia de los algoritmos tradicionales basados en RL, el agente DRL propuesto para desarrollar este mecanismo AC incluye una *Online NN*, una *Target NN* y un *Buffer* de Reproducción de Experiencia para maximizar la recompensa, obtener un aprendizaje acelerado y mejorar la estabilidad del proceso de aprendizaje (consulte la sección 4.2.4 y 4.2.5). Con el fin de lograr el adecuado acoplamiento entre el agente DRL y su entorno, se modela el espacio de estados, espacio de acciones y

ecuaciones de recompensa, los cuales rigen las políticas de toma de decisiones y el proceso de aprendizaje.

4.2.1 Espacio de Estados

El espacio de estados \mathcal{S} se define como el conjunto de estados que el agente DRL puede observar [100]. Cada estado $s \in \mathcal{S}$ se identifica mediante una tupla de 8 elementos (ver Ecuación 4.3), cuatro de ellos representan potenciales de satisfacción de cada tipo de SLR (Ecuación 4.2) y los cuatro siguientes indican el número de SLR solicitadas por cada tipo durante la ventana de tiempo. El potencial de satisfacción es una métrica establecida mediante experimentación, calculado por el Módulo de Monitoreo para identificar la capacidad de la red para satisfacer los requisitos de un tipo específico de SLR (RS, URLLC, eMBB y mMTC). El potencial de satisfacción hace uso del parámetro β_n , con el fin de controlar el impacto que presenta cada tipo de recurso (CPU_{borde} , $CPU_{central}$ y BW) sobre el potencial de cada tipo de SLR. El Módulo de Monitoreo usa la información del porcentaje de disponibilidad de CPU de los nodos (nodos de borde y nodos centrales) y el ancho de banda de los enlaces que componen cada *slice* en el grupo *Slices Usables* para calcular los potenciales de satisfacción.

$$P_{Type} = \beta_1 \cdot CPU_{borde} + \beta_2 \cdot CPU_{central} + \beta_3 \cdot BW \quad (4.2)$$

Como el valor de los potenciales se encuentra dentro de un rango $[0,1]$, el mecanismo considera un rango continuo que permite al agente DRL observar de forma granular todos los estados posibles en los que podría encontrarse la red. El mecanismo no establece estados finitos porque eso limita el espacio de búsqueda. Por lo tanto, cada estado se define de la siguiente manera:

$$\mathcal{S} = \{P_{RS}, P_{URLLC}, P_{eMBB}, P_{mMTC}, \eta_1, \eta_2, \eta_3, \eta_4\} \quad (4.3)$$

Donde:

P_{RS} : Potencial de satisfacción para SLR tipo RS

P_{URLLC} : Potencial de satisfacción para SLR tipo URLLC

P_{eMBB} : Potencial de satisfacción para SLR tipo EMBB

P_{mMTC} : Potencial de satisfacción para SLR tipo mMTC

η_1 : número de SLR tipo RS

η_2 : número de SLR tipo URLLC

η_3 : número de SLR tipo eMBB

η_4 : número de SLR tipo mMTC

4.2.2 Espacio de Acciones

Dado que el agente DRL busca encontrar un conjunto efectivo de SLR para satisfacer los requisitos de latencia de *slices* para cirugía remota y optimizar el lucro (ver 4.2.3); el espacio de acción \mathcal{A} se compone del porcentaje de las SLR de cada tipo que deben aceptarse. En cada paso de tiempo, el agente selecciona una acción $a \in \mathcal{A}$ denotada como una tupla $a = \{\Omega_{RS}, \Omega_{URLLC}, \Omega_{eMBB}, \Omega_{mMTC}\}$, buscando cumplir con los valores de latencia y optimizar el parámetro lucro. Una muestra puede ser la acción $\{0.5, 0.3, 0.1, 0.4\}$, donde el 50%, 30%, 10% y 40% del total de peticiones recibidas de RS, URLLC, mMTC y eMBB son pre-admitidas, respectivamente.

4.2.3 Función de Recompensa

En cada estado, el agente DRL elige una acción que afecta el entorno y tendrá como resultado una recompensa. Esta recompensa se encuentra directamente relacionada con el objetivo del mecanismo AC; satisfacer la latencia de los *slices* (especialmente los servicios RS y URLLC) y optimizar el lucro del InP. Por lo tanto, la recompensa total propuesta (R) consta de una recompensa relacionada con la latencia ($R_{latencia}$)

y una recompensa relacionada con el lucro (R_{pr}) (ver ecuación 4.4). Ambas recompensas están acompañadas de factores denominados δ , los cuales permiten controlar el impacto de R_{pr} y $R_{latencia}$ en la recompensa total R .

$$R = \delta_1 \cdot R_{latencia} + \delta_2 \cdot R_{pr} \quad (4.4)$$

Recompensa de Latencia: El cálculo de $R_{latencia}$ se realiza considerando el término L_{di} (ver Ecuación 4.5), el cual se calcula como la diferencia entre la latencia obtenida por la SLR desplegada (L_i) y la latencia requerida por dicha SLR (L_r). Esta diferencia nos permite obtener valores positivos cuando satisface la latencia y valores negativos en caso contrario. L_{di} se inserta en una función tangente hiperbólica (ver Ecuación 4.6), que está acotada entre los valores $[-1,1]$. Cuando la latencia obtenida por las SLR desplegadas tiende al valor mínimo posible, se genera una R_{slr_i} cercana a 1, mientras que cuando dichas latencias son altas (incumpliendo la latencia requerida), R_{slr_i} se acerca a -1.

$$L_{di} = L_r - L_i \quad (4.5)$$

$$R_{slr_i} = \frac{e^{L_{di}} - e^{-L_{di}}}{e^{L_{di}} + e^{-L_{di}}} \quad (4.6)$$

Donde:

L_i : latencia alcanzada por un *slice*.

L_r : latencia requerida por una SLR específica.

L_{di} : diferencia entre latencia requerida y latencia alcanzada.

R_{slr_i} : recompensa de cada SLR desplegada con respecto a la latencia alcanzada $[1, -1]$.

Dado que este mecanismo requiere analizar el desempeño de cada tipo de SLR, las múltiples recompensas R_{slr_i} obtenidas se clasifican según su tipo (RS, URLLC,

eMBB y mMTC) y se calcula un total de 4 promedios R_{tipo} mediante la Ecuación 4.7.

$$R_{tipo} = \frac{\sum_{i=0}^{N_{tipo}} R_{slr_i}}{N_{tipo}} \quad (4.7)$$

Donde:

N_{tipo} : número de slices aceptados por tipo de SLR.

R_{tipo} : recompensa promedio del grupo de SLR admitidas por tipo de SLR.

El resultado de R_{tipo} representa el promedio de las recompensas R_{slr_i} generadas por un tipo específico de SLR (R_{RS} , R_{URLLC} , R_{eMBB} y R_{mMTC}). Estos promedios se utilizan para calcular $R_{latencia}$, como se evidencia en la Ecuación 4.8. Dado que este mecanismo busca satisfacer la latencia de la cirugía remota, se establecen factores α para controlar el impacto que tendría este servicio en la recompensa por latencia, lo que nos permite dar especial prioridad a RS y URLLC (sección de evaluación 5).

$$R_{latencia} = \alpha_1 \cdot R_{RS} + \alpha_2 \cdot R_{URLLC} + \alpha_3 \cdot R_{eMBB} + \alpha_4 \cdot R_{mMTC} \quad (4.8)$$

Recompensa de Lucro: R_{pr} es la recompensa relacionada con el lucro obtenida por el InP después de cada acción del agente, el cual se define como la diferencia entre los ingresos generados por el grupo de SLR desplegadas (Re) y los costos de implementación (C) relacionados con los recursos utilizados por los SLR desplegadas. Los ingresos pagados por el *tenant* son válidos solo si el segmento se implementó correctamente (en términos de ancho de banda, CPU y latencia), en caso contrario, se genera una penalización (F) equivalente a los C . El costo de implementación y el lucro considerado en un solo *slice* se describen de acuerdo con las ecuaciones 4.9 y 4.10, respectivamente.

$$C_i = \sum_{n=0}^k bw(link_n) \cdot G_{bw} + \sum_{m=0}^j cpu(vn.f_m) \cdot G_{cpu} \quad (4.9)$$

$$pr(slr_i) = (Re_i - C_i) \cdot T_i \quad (4.10)$$

Donde:

$pr(slr_i)$: lucro por implementar SLR_i en un *slice*.

Re_i : ingresos generado por el despliegue de una SLR_i en un *slice*.

C_i : costo de implementación (en términos de recursos) por aceptar SLR_i en el *slice*.

T_i : tiempo de vida de la SLR_i en el *slice*.

k : número de enlaces para implementar la SLR_i .

j : número de vnfs para implementar la SLR_i .

G_{bw} : costo por unidad de ancho de banda.

G_{cpu} : costo por unidad de CPU.

De esta forma, las R_{pr} obtenidas del grupo de SLR desplegadas se calcula de acuerdo con la siguiente ecuación:

$$R_{pr} = \frac{\sum_{i=0}^p pr(slr_i) - \sum_{i=0}^j F_i}{pr_{max} \cdot T_{max}} \quad (4.11)$$

Donde:

pr_{max} : beneficio máximo que se podría generar haciendo uso de todos los recursos disponibles.

T_{max} : tiempo de duración del episodio.

F_i : penalización generada por incumplimiento de la latencia.

p : número de SLR desplegadas correctamente.

j : número de *slices* que no cumplieron con la latencia requerida.

Tanto $R_{latency}$ como R_{pr} están en rangos comprendidos entre $[-1,1]$, lo que facilita su suma, análisis y cálculo para el entrenamiento del agente DRL.

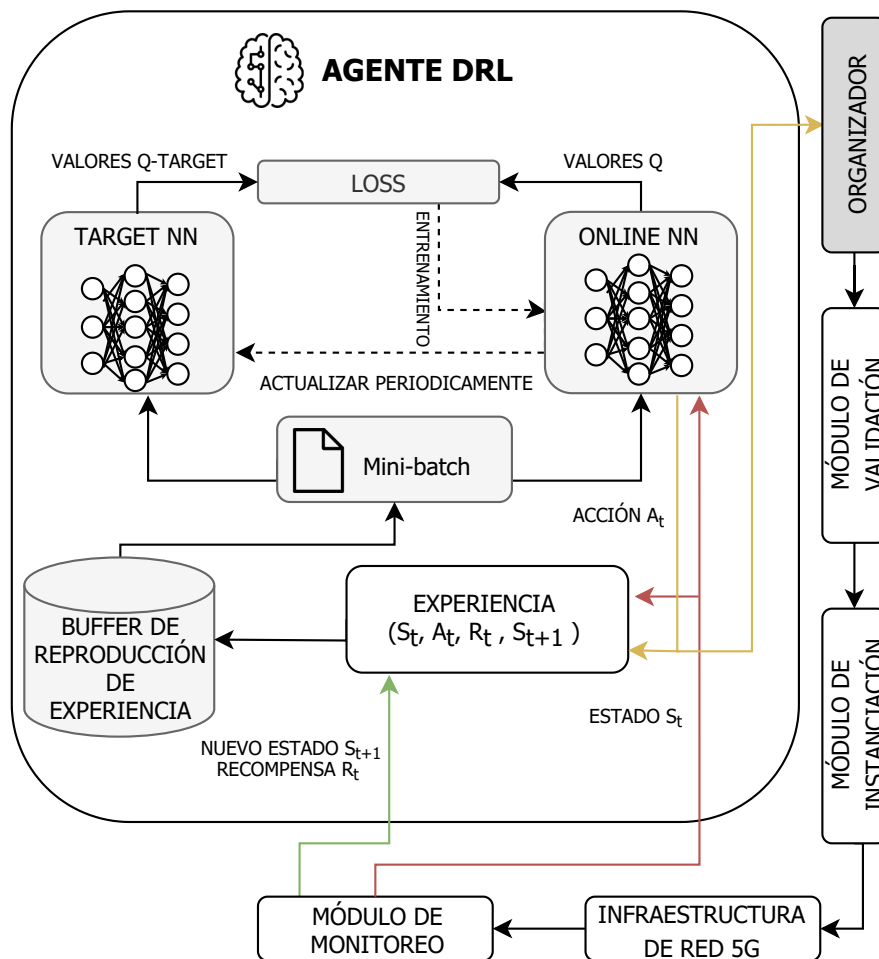


Figura 4.4: Arquitectura interna del agente deep reinforcement learning

4.2.4 Redes Neuronales

Los algoritmos tradicionales de RL se basan en una representación tabular para almacenar los valores Q de las acciones para cada estado [57]. Sin embargo, para acelerar el proceso de aprendizaje y mejorar la generalización de los estados del agente, el agente DRL utiliza una función de aproximación basada en una o varias NN (*Neural Networks*) denominada DQN [101], lo cual le permite reducir la carga computacional, ampliar el espacio de estados y mejorar la capacidad predictiva del agente. En la arquitectura propuesta para el agente (ver Figura 4.4) se utilizan dos NN, *Online NN* y *Target NN*. El agente utiliza la *Online NN* para calcular los valores de Q en los estados actuales $S_t(Q(s_t, \mathcal{A}_t))$. Esta NN se entrena en cada paso de tiempo mediante un método típico de aprendizaje supervisado, que busca reducir la función de pérdida (ver Ecuación 4.13). Este entrenamiento permite actualizar los pesos usando el enfoque *Gradient Descent* y el algoritmo *Backpropagation*.

$$Q^+(S_t, \mathcal{A}_t) = R_t + \gamma \cdot \max_{\mathcal{A}} Q(S_{t+1}, \mathcal{A}_t) \quad (4.12)$$

$$Loss = (Q^+(S_t, \mathcal{A}_t) - Q(S_t, \mathcal{A}_t))^2 \quad (4.13)$$

Target NN estima los valores Q-*Target* (Ver la ecuación 4.12), que se utilizan para calcular la función de pérdida *Loss*, la cual facilita el proceso el entrenamiento de la *Online NN*. La función de pérdida se basa en el MSE (*Mean Squared Error*) [102, 103], donde las ponderaciones *Target NN* se congelan temporalmente y se utilizan como término objetivo, mejorando la estabilidad del proceso de aprendizaje del agente. Cabe señalar que los pesos de la *Online NN* se transfieren frecuentemente a la *Target NN* cada τ pasos de tiempo para que las estimaciones de la *Target NN* se actualicen y el agente DRL mejore su comportamiento respecto al objetivo de este mecanismo de AC.

En la arquitectura de una *Target NN* y la *Online NN*, las neuronas de la capa de entrada se establecen según el número de variables que componen cada estado s (i.e. una neurona de entrada por variable en la tupla del espacio de estados) y las

neuronas de la capa de salida según el número de acciones posibles a que el agente podría tomar en un estado (una neurona de salida por cada acción posible) [104]. Teniendo en cuenta lo anterior, tanto la *Online* NN como la *Target* NN requieren un número de 8 neuronas en sus capas de entrada debido a las variables definidas en la sección de 4.2.1. Asimismo, sus capas de salida utilizan n neuronas, definidas mediante experimentación, donde cada neurona calcula el valor Q_i para cada acción $a_i \in \mathcal{A}$. Cabe señalar que el número de capas ocultas y las neuronas en ambas redes serán variadas en la sección 5 con el fin de encontrar la configuración conducente a mejores resultados.

4.2.5 Reproducción de Experiencia

El agente DRL captura muestras de experiencias previas (s_t, a_t, R_t, s_{t+1}) y las almacena en un Buffer de Reproducción de Experiencias (B) de tamaño limitado. En cada iteración de entrenamiento, el agente de DRL selecciona un *Mini-Batch* aleatorio de experiencias para entrenar la *Online* NN, haciendo uso del enfoque de *Gradient Descent* [105]. El uso de *Mini-Batches* para el proceso de aprendizaje disminuye los tiempos de convergencia [97] [100], la varianza de aprendizaje [97] [100] y la correlación entre muestras consecutivas [106]. El tamaño del *Mini-Batch* se establece mediante un proceso de refinamiento, el cual permite encontrar la combinación adecuada para el equilibrio entre la satisfacción de la latencia y la optimización del lucro.

4.2.6 Exploración y Explotación

Todo agente DRL requiere un método de exploración adecuado para potenciar su proceso de aprendizaje y mejorar sus resultados a medida que toma decisiones. En este trabajo de grado, el agente DRL hace uso del método *Decay ϵ - greedy* [107], el cual busca el equilibrio adecuado entre la exploración (acciones aleatorias para descubrir nuevas recompensas óptimas) y la explotación (acciones encontradas por el agente para maximizar las recompensas). En este orden de ideas, el parámetro $\epsilon \in [0, 1]$, determina la probabilidad de exploración del agente. Inicialmente, se

establece un valor máximo de ϵ_{max} , configurado para disminuir según una función lineal con factor de decremento de pendiente (df) hasta llegar a un ϵ_{min} , donde la probabilidad de exploración llega a su valor mínimo. De esta manera, de acuerdo con la Ecuación 4.14, el agente reduce su probabilidad de exploración mientras aumenta su probabilidad de explotación.

$$\epsilon_t = \epsilon_{max} - (nsteps_t \times df) \quad (4.14)$$

En cada paso de tiempo, el agente genera un valor aleatorio $rv \in [0, 1]$ que indicará si el agente debe explorar una acción aleatoria o, por el contrario, explotar aquella con mayor recompensa (ver Ecuación 4.15).

$$a = \begin{cases} \max_{\mathcal{A}} Q_t(s_t, a), & \text{if } rv > \epsilon_t \\ \text{random action}, & \text{otherwise} \end{cases} \quad (4.15)$$

4.3 Algoritmo de Control de Admisión Basado en Aprendizaje por Refuerzo Profundo Enfocado en el Servicio de Cirugía Remota

El Algoritmo 1 presenta un mecanismo AC basado en DRL para aceptar el conjunto de SLR que permite un equilibrio adecuado entre la satisfacción de latencia de SLR (especialmente aquellas enfocadas en los tipos RS y URLLC) y la optimización del lucro. El algoritmo recibe los siguientes parámetros como entrada: espacio de estados \mathcal{S} , espacio de acciones \mathcal{A} , número de episodios de aprendizaje n , la máxima (ϵ_{max}) y mínima (ϵ_{min}) probabilidad de exploración, factor de descuento γ , factor de decremento df y frecuencia de actualización τ para la *Target* NN. La salida es el conjunto de SLR admitidas que contribuyen al objetivo del mecanismo AC, alcanzar un equilibrio adecuado entre optimizar el lucro y satisfacer la latencia.

El Algoritmo 1 comienza inicializando la *Online* NN, la *Target* NN, el Búffer de

Reproducción de Experiencias B y el Repositorio de *slices*. El Algoritmo AC usa un bucle externo (línea 2 a línea 35) de n episodios de aprendizaje, donde en primera medida el agente DRL observa el estado inicial de la red s_t (línea 3). Cada episodio describe un conjunto de pasos de tiempo y en cada uno de ellos, el agente observa el estado actual del entorno, toma una acción y recibe una recompensa. La acción tomada por agente altera el estado del entorno, llevando al agente a un nuevo estado. Este conjunto de pasos siguen un bucle interno (línea 4 a línea 33), que concluye en un estado final, el cual se presenta cuando transcurren un número finito de m pasos de tiempo.

El ciclo interno (línea 4 a línea 33) comienza a actualizar el parámetro de exploración ϵ mediante la ecuación 4.14 (línea 5). El Módulo de Instanciación libera los recursos asociados a los *slices* que han completado su tiempo de vida (línea 6) y grupo *Slices Usables* se actualiza (línea 7), para observar un estado real de la red antes de cualquier decisión del agente DRL. Luego, el agente de DRL selecciona una acción a_t (de acuerdo con la ecuación 4.15), que contiene la tupla de porcentaje de aceptación de cada tipo de SLR (línea 8). El Organizador utiliza la tupla de porcentaje para seleccionar el conjunto de SLR pre-admitidas y clasificarlas según su tipo en el grupo SLR Organizadas, donde RS y URLLC (línea 9) se procesan antes que los otros tipos de SLR. Cada *slr* en el grupo SLR Organizadas se envía al Sub-módulo de verificación, que verifica la disponibilidad de recursos para satisfacer los QoS del *slr* (línea 11). Si el *slr* es factible para ser desplegado, se envía al Sub-módulo de Reservación que reserva los recursos de red necesarios (línea 13); de lo contrario, *slr* se rechaza y el InP recibe una penalización (línea 17). Luego, el Módulo de Instanciación asigna un tiempo de vida a los recursos reservados para el *slr*, los despliega y almacena su información dentro del grupo *Slices Desplegados* (línea 14).

El agente obtiene una recompensa R_t (línea 20) por el conjunto de SLR admitidas (de acuerdo con la ecuación 4.4) y luego, se genera el siguiente estado s_{t+1} (línea 21). La tupla de experiencia (s_t, a_t, R_t, s_{t+1}) recibida de la decisión del agente se almacena en B (línea 22). El agente selecciona un *mini-batch* aleatorio de experiencias almacenadas en B para entrenar a la *Online NN* (línea 23). El valor $Q(s_t, a_t)$ se estima usando la *Online NN* (línea 24) y mediante la *Target NN* se calcula el $Q^+(s_t, a_t)$, que según la definición teórica tiene un valor de R_t cuando el siguiente estado es el

estado final (línea 26) [57] o de lo contrario un valor de $R_t + \gamma \cdot \max Q^+(s_{t+1}, a)$. Con los valores obtenidos previamente, la función de pérdida se calcula usando la ecuación 4.13, que permite al agente actualizar los pesos θ usando el enfoque de *Gradient Descent* y el algoritmo *Backpropagation* (línea 31). Al final de cada paso de tiempo, se realiza una actualización donde el siguiente estado s_{t+1} (observado en la línea 21) se convierte en el estado actual s_t . La *Target NN* se actualiza con una frecuencia de τ pasos de tiempo para evitar el uso de información antigua, lo que permite la mejora progresiva del agente DRL (línea 34). A partir de este momento el algoritmo del mecanismo de AC comienza un nuevo episodio.

Algoritmo 1: Algoritmo de control de admisión basado en aprendizaje por refuerzo profundo

Entrada : Número de episodios de aprendizaje: n
 Parámetros de exploración: $\epsilon_{max}, \epsilon_{min}$
 Factor de descuento: γ
 Factor de decremento: df
 Frecuencia de actualización : τ

Resultado: SLR que optimizan la satisfacción de la latencia y el lucro

```

1 Se inicializa: Repositorio, Online NN  $Q$ , Target NN  $Q^+$  y el buffer de reproducción de experiencia  $B$ 
2 for episode  $\leftarrow 1$  to  $n$  do
3     El agente observa el estado inicial  $s_t$  ;
4     while siguiente estado  $s_{t+1}$  no sea el estado final do
5         Se actualiza la variable  $\epsilon$  mediante la ecuación 4.14;
6         Módulo de Instanciación libera recursos de los SLR que han completado su tiempo de vida;
7         Se actualiza el grupo Slices Usables;
8         El agente selecciona una acción  $a_t$  de acuerdo a la ecuación 4.15;
9         El agente invoca al Organizador para organizar SLR y priorizar RS y URLLC dentro de la lista SLR
            organizadas;
10        for  $slr \in SLROrganizadas$  SLR do
11            El Organizador envía la  $slr$  al Submódulo de verificación, donde verifica su cumplimiento con la
                latencia ;
12            if  $slr$  puede ser desplegado then
13                El Módulo de reservación reserva los recursos requeridos por la  $slr$ .;
14                El Módulo de Instanciación asigna el tiempo de vida de los recursos requeridos por la  $slr$  y
                    los almacena en el grupo Slices Desplegados;
15            end
16            else
17                La  $slr$  será rechazada;
18            end
19        end
20        El agente obtiene una recompensa  $R_t$  mediante la ecuación 4.4;
21        El agente observa el siguiente estado  $s_{t+1}$ ;
22        Almacena la experiencia  $e_t = (s_t, a_t, R_t, s_{t+1})$  dentro de  $B$ ;
23        Se selecciona un mini-batch aleatorio de experiencias en  $B$ ;
24        Se estima  $Q(s_t, a_t)$  con la Online NN;
25        if Siguiente estado  $s_{t+1}$  es el estado final then
26             $Q^+(s_t, a_t) = R_t$  ;
27        end
28        else
29             $Q^+(s_t, a_t) = R_t + \gamma \cdot \max Q^+(s_{t+1}, a)$ 
30        end
31        Se minimiza la función de pérdida (Ver ecuación 4.13) mediante gradiente descendente y actualiza
            los pesos  $\theta$  de la Online NN  $Q$ ;
32        Se actualiza el estado actual  $s_t \leftarrow s_{t+1}$ ;
33    end
34    Cada  $\tau$  episodios, el agente copia los hiperparámetros desde la Online NN hacia Target NN ( $Q^+ = Q$ )
35 end
    
```

Capítulo 5

Evaluación y Análisis de Resultados

5.1 Evaluación

En esta sección presentamos la evaluación del mecanismo de AC, con el fin de observar su eficiencia en términos de satisfacción de la latencia y el lucro. La subsección 5.1.1 presenta el entorno de pruebas y los requisitos de QoS de cada tipo de SLR. La subsección 5.1.2 define las métricas de evaluación. La sección 5.1.3 presenta los resultados de las variaciones de los parámetros e hiperparámetros propuestos en la sección 4.2. La subsección 5.2 presenta resultados y análisis entre el mecanismo de AC de *slices* de red diseñado para este trabajo de grado y propuestas similares. Adicionalmente, se presentan resultados obtenidos mediante las variaciones en la topología de la red.

5.1.1 Ambiente de Experimentación

El ambiente de experimentación establece requisitos específicos de QoS en términos de latencia en el núcleo de la red para cada tipo de SLR. El servicio de cirugía remota presenta una demanda de latencia estricta, comprendida en un rango entre

	BW (unidades)	CPU (unidades)	Latencia en el núcleo (ms)	Tiempo de Vida (λ)
RS	4	8	0.1	30
URLLC	3	5	0.3	30
eMBB	2	4	0.8	15
mMTC	1	4	1	8

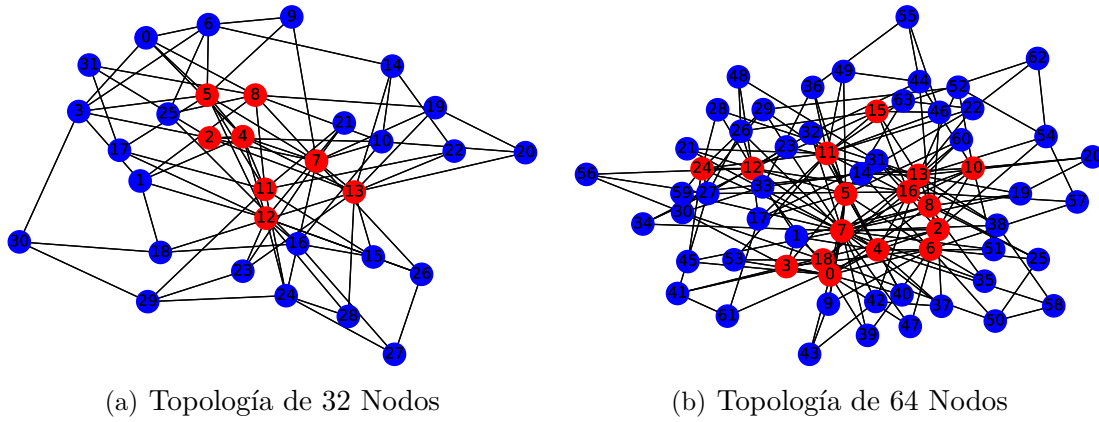
Tabla 5.1: Requerimientos de calidad de servicio para los *Slices Request*

Figura 5.1: Topologías de 32 y 64 nodos

[0.1,0.5] ms [7][11]. Por lo tanto, RS y URLLC se han establecido con requisitos de latencia de 0,1 y 0,3 ms, respectivamente. Además, se establecen latencias de 0,8 y 1 ms en los servicios eMBB [108, 109] y mMTC [110], respectivamente. Además, los parámetros de ancho de banda, CPU y tiempo de vida son definidos para cada tipo de SLR (ver Tabla 5.1), con el fin de simular el despliegue de funciones de red en la infraestructura 5G. El tiempo de vida denota el tiempo que cada SLR está activa en la red, el cual se establece mediante un proceso de *Poisson* con media λ [99].

El mecanismo se evaluó en topologías de 32 y 64 nodos de NFVI-PoPs (ver figura 5.1), las cuales se crean a partir del Modelo Barabási-Albert¹ [111]. En cada uno de ellos se tienen en cuenta los mismos criterios; El 75% de los nodos pertenecen a nodos de baja capacidad (100 unidades de CPU disponibles por nodo), el 25%

¹La teoría de redes define el Modelo de Barabási-Albert como un algoritmo empleado para generar redes aleatorias complejas con un distribución de grado potencial y libres de escala. Las topologías de red generadas por este modelo son utilizadas en sistemas de internet, el *world wide web*, algunas redes sociales y redes eléctricas [111].

restante son nodos de alta capacidad (400 unidades de CPU disponibles por nodo) y enlaces virtuales con 100 unidades de ancho de banda disponibles. Los procesos de construcción, simulación y evaluación de los distintos componentes de este mecanismo se desarrollaron a través del lenguaje de programación Python 3 (Ver figura 5.2). Estos procesos se ejecutaron en una VM (*Virtual Machine*) Ubuntu 16.04 con 8 GB de RAM, que se aloja en un host con CPU Intel Core i5-6200U. Dado que la evaluación de este mecanismo de AC se lleva a cabo en un entorno simulado, la duración de todos los procesos se representa mediante unidades de tiempo discretas (es decir, $t = 1,2,3$, etc.). La creación, implementación, monitorización y despliegue de múltiples SLR, incluida la infraestructura propuesta, se simuló utilizando la biblioteca NetworkX [112].

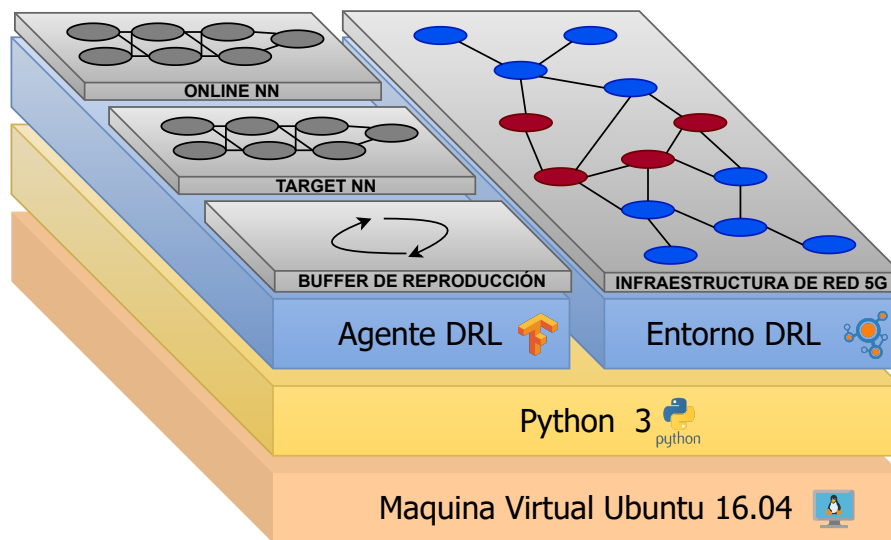


Figura 5.2: Arquitectura del ambiente de experimentación

El prototipo del mecanismo de control de admisión se basa en 4 archivos python fundamentales: **main**, **request**, **DqnAgent** y **env** (Ver figura 5.3). El archivo **request** contiene la información de los tipos de SLR diseñados para nuestro mecanismo, sus características de QoS y su distribución topológica. Estas SLR se invocan desde el archivo **env**, el cual reúne las funcionalidades de los módulos Monitoreo, repositorio, reconocimiento, validación e instanciación. De esta manera, el archivo **env** contiene toda la información del entorno DRL, el cual contempla el modelado de la

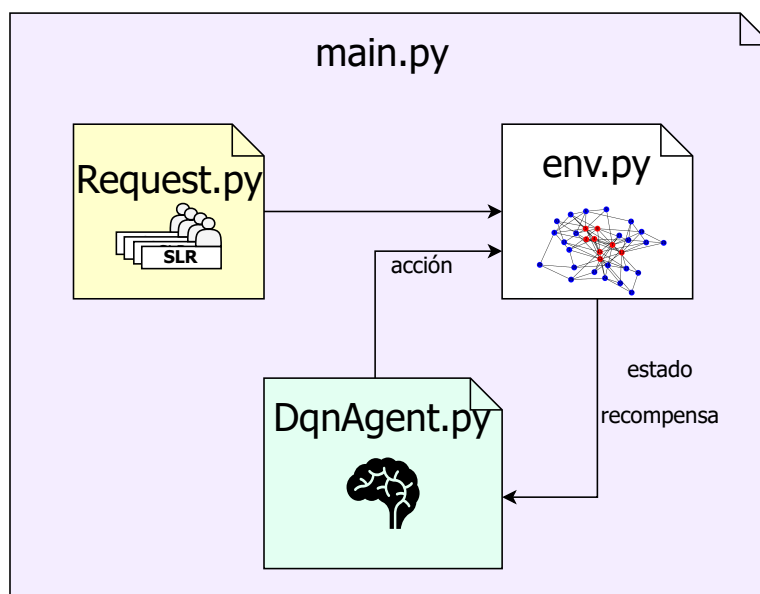


Figura 5.3: interacción de archivos python

topologías de red, espacio de estados, espacio de acciones y función de recompensa. Esta información se traslada continuamente al archivo **DqnAgent**, el cual establece la lógica del agente DRL y sus componentes. El agente DRL toma acciones que son enviadas al **env**, afectando el estado actual del entorno DRL. La comunicación entre los elementos previamente descritos (**request**, **DqnAgent** y **env**) se establece a través del archivo central denominado **main**, que modela el ciclo de episodios y conecta las tareas de cada archivo dentro de múltiples pasos de tiempo. Asegurar una correcta interacción entre los archivos, permite el correcto funcionamiento del prototipo de control de admisión y el desarrollo de un ambiente de experimentación robusto.

5.1.2 Métricas

En este trabajo de grado se establecen dos métricas esenciales que nos ayudan a determinar el desempeño de este mecanismo. El primero es *SaR* (*Satisfaction Ratio*), que permite observar qué porcentaje del conjunto de las SLR aceptadas por el mecanismo AC se satisfacen en términos de latencia, durante un episodio. Esta

métrica se expresa matemáticamente de la siguiente manera:

$$SaR = \frac{\sum_{i=0}^n slr_i}{\sum_{j=0}^m slr_j} \quad (5.1)$$

Donde:

n : Número de SLR satisfechas en términos de latencia.

m : Número de SLR aceptadas por el Módulo de AC en un episodio.

Adicionalmente, consideramos una métrica de lucro, para visualizar las ganancias monetarias del InP (ver Ecuación 4.11). Para representar el resultado monetario del agente, se establecieron costos de 3, 2 y 1 unidades monetarias por cada unidad de CPU_{Borde} , $CPU_{núcleo}$ y BW_{Link} , respectivamente.

5.1.3 Configuración de los Experimentos

Como se puede ver en la Sección 4.2, el mecanismo AC propuesto, comprende una gran cantidad de parámetros en las múltiples ecuaciones de estado, recompensa y en toda la estructura del agente DRL. Por lo tanto, para lograr el máximo rendimiento de este mecanismo AC, se realizó un proceso de refinamiento para encontrar el conjunto de parámetros y la combinación adecuada que logren optimizar el rendimiento de las métricas propuestas (ver subsección 5.1.2). Este proceso de refinamiento se realizó en la topología de 32 nodos, ya que representa el escenario con mayor limitación en recursos de red y por lo tanto, revela el comportamiento de este mecanismo cuando se pone a prueba sobre la situación más exigente.

Inicialmente, el parámetro β_n se varió en la ecuación 4.2. Se proponen cinco opciones para observar el comportamiento del potencial de satisfacción para cada tipo de SLR (ver Tabla 5.2). Las figuras 5.4(a), 5.4(b), 5.4(c) y 5.4(d) revelan un patrón común entre los diferentes tipos de SLR, donde los recursos disponibles en los nodos

Opción	RS - URLLC - mMTC			eMBB	
	β_1	β_2	β_3	β_2	β_3
A	0.15	0.75	0.1	0.1	0.9
B	0.65	0.1	0.25	0.9	0.1
C	0.45	0.45	0.1	0.5	0.5
D	0.25	0.5	0.25	0.7	0.3
E	0.1	0.1	0.8	0.3	0.7

Tabla 5.2: Variación de β

Opción \ α_n	α_1	α_2	α_3	α_4
A	0.6	0.2	0.1	0.1
B	0.4	0.3	0.15	0.15
C	0.25	0.25	0.25	0.25
D	0.15	0.15	0.4	0.3
E	0.1	0.1	0.6	0.2
F	0.4	0.4	0.1	0.1

Tabla 5.3: Variación de α

centrales representan una gran importancia al determinar si un servicio se puede implementar correctamente. Esto se debe a que los nodos centrales son consumidos por todos los tipos de SLR y se utilizan con gran frecuencia, a diferencia de los nodos de borde. Demostrando así que la opción D tiene el *SaR* más alto para los servicios RS y URLLC, mientras que las opciones A y D favorecen a mMTC y eMBB, respectivamente.

Luego, se varía el parámetro α_n . Cabe señalar que este parámetro controla el impacto que tiene cada servicio en $R_{latencia}$ (consulte la ecuación 4.8). Las figuras 5.5(a), 5.5(b), 5.5(c) y 5.5(d) muestran que la tasa de satisfacción de cada tipo de SLR es directamente proporcional a su factor correspondiente α_n . Dado que el mecanismo está enfocado principalmente en cumplir con los estrictos requisitos de latencia de RS y URLLC, se busca la configuración que en conjunto proporcione el *SaR* más elevado para estos servicios. La opción A representa un mecanismo AC centrado en la satisfacción del servicio RS con un *SaR* = 96%, sin embargo, esta opción perjudica el desempeño del servicio URLLC, que se reduce a un *SaR* de 86%, lo que afecta el objetivo de esta investigación. Por otro lado, la opción F

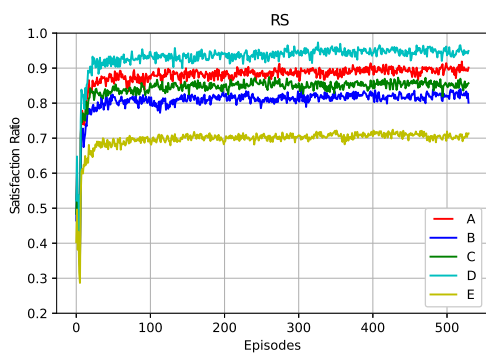
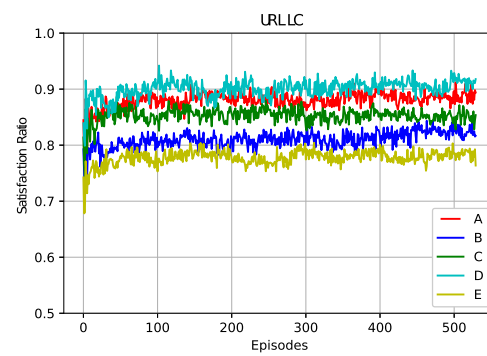
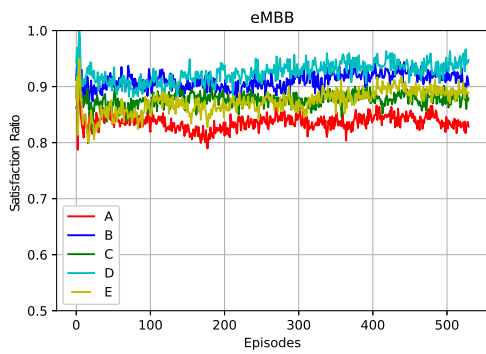
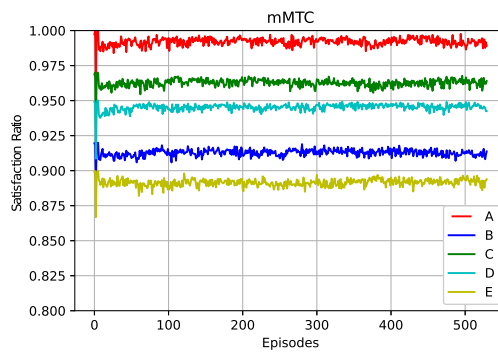
(a) Variación P_{RS} (b) Variación P_{URLLC} (c) Variación P_{EMBB} (d) Variación P_{MMTC}

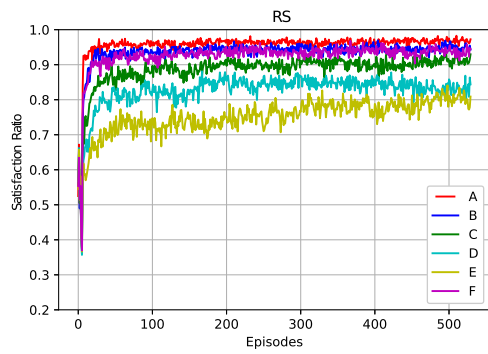
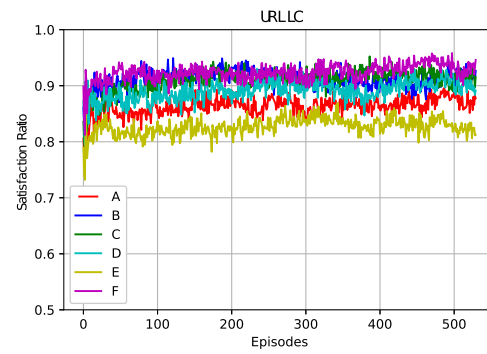
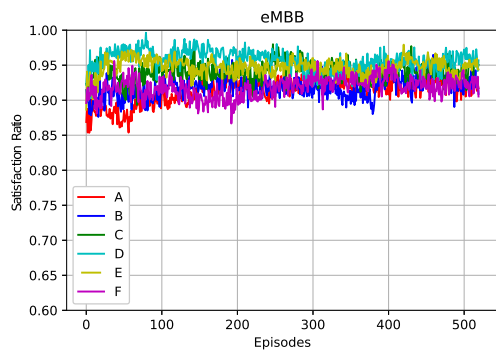
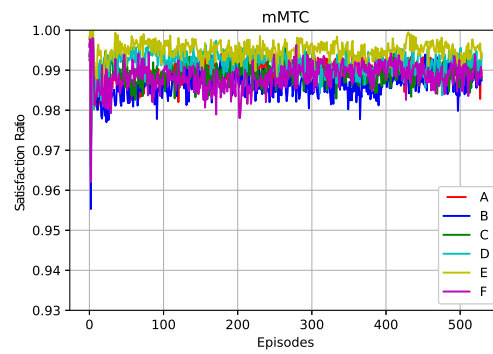
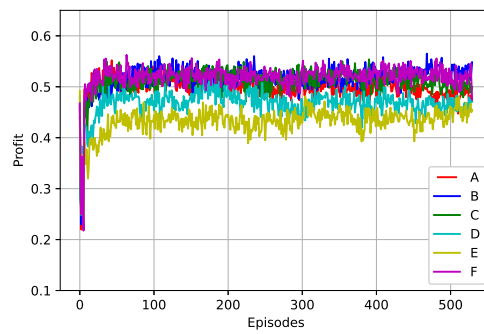
Figura 5.4: Variación de potenciales de satisfacción

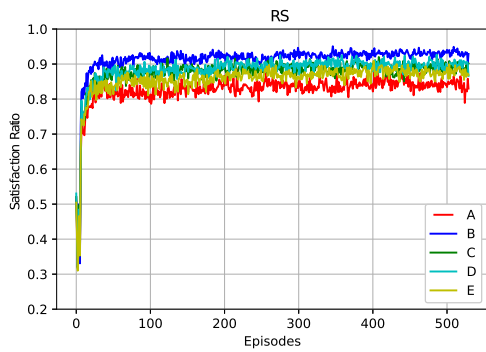
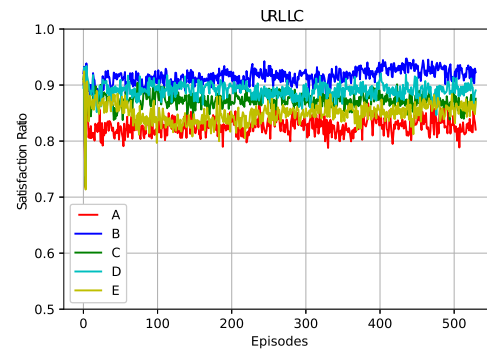
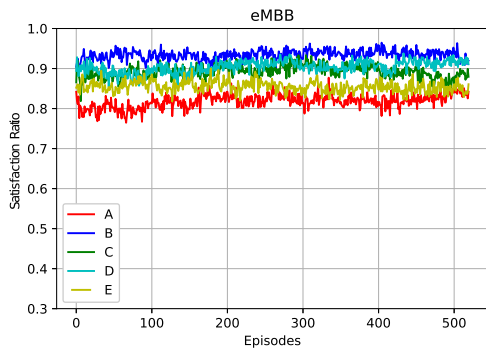
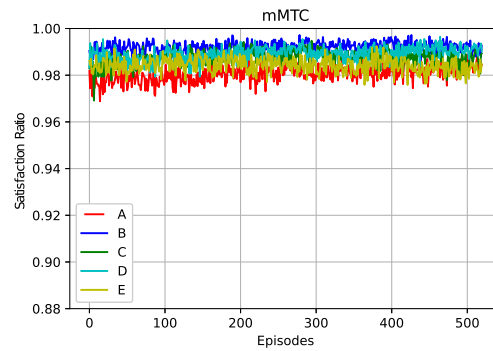
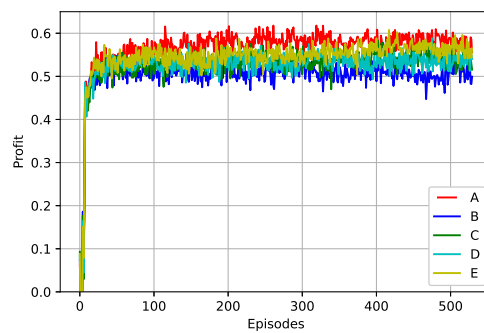
Opción \ δ_n	δ_1	δ_2
A	0.2	0.8
B	0.8	0.2
C	0.5	0.5
D	0.6	0.4
E	0.4	0.6

Tabla 5.4: Variación de δ

presenta valores de $\alpha = 0.4$ para los servicios RS y URLLC, lo que refleja su impacto significativo en las decisiones del mecanismo AC. Esto permite alcanzar valores de SaR de hasta 93% para RS y URLLC, siendo la opción con mayor SaR logrado por el mecanismo AC. Cabe resaltar que los servicios eMBB y mMTC no requieren estrictos requisitos de latencia, de manera que presentan menor dificultad para ser satisfechos en comparación con RS y URLLC. Por lo tanto, sus valores de SaR no se ven afectados sustancialmente por la priorización de los servicios RS y URLLC, logrando así valores de SaR de 93% (eMBB) y 99% (mMTC). Al mismo tiempo, la figura 5.5(e) muestra que el lucro disminuye notablemente cuando se priorizan las SLR de tipo eMBB y mMTC en las opciones D y E; esto se debe a que si bien los SaR de ambos tipos alcanzan su valor máximo, este tipo de servicios no presentan un aporte lucrativo considerable al InP, en contraste con el servicio de cirugía remota. Esto refuerza que la opción F sigue contribuyendo positivamente al objetivo de este trabajo de grado.

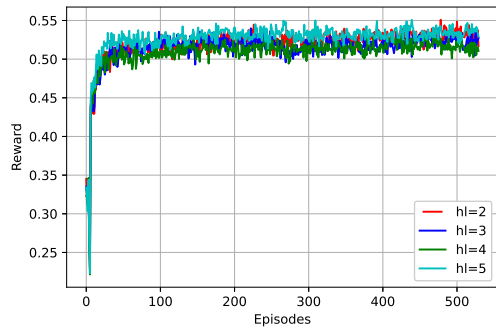
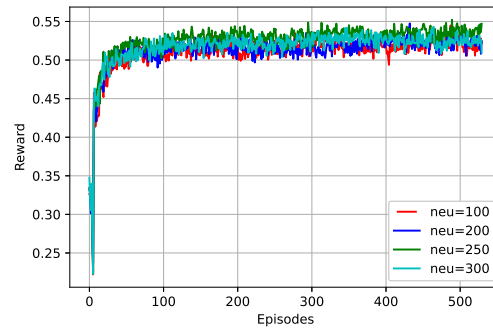
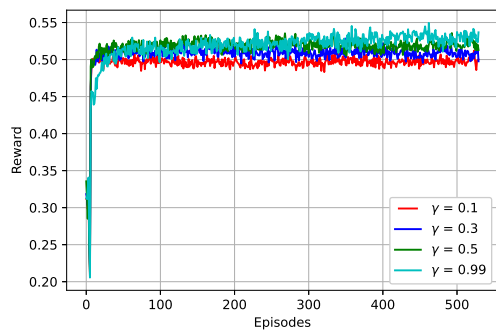
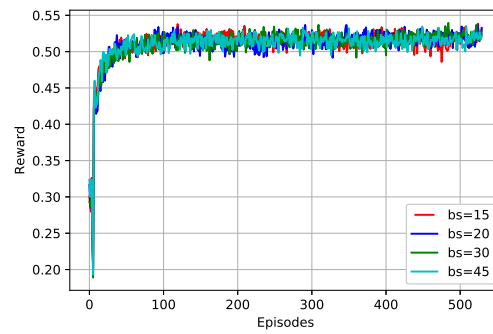
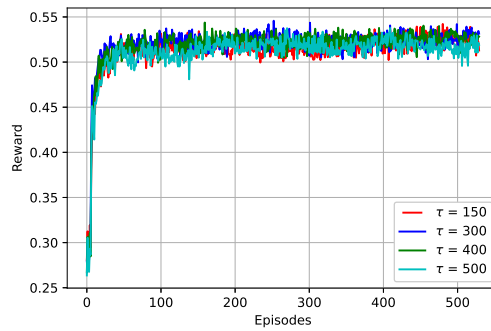
Para el cálculo de los parámetros δ_n en la ecuación de la recompensa total (ver ecuación 4.4), se realiza un proceso de refinamiento que contempla 5 opciones. En cada una de estas opciones se varió el impacto de $R_{latencia}$ y R_{pr} sobre la recompensa total, lo que permite determinar la combinación con mejor desempeño en términos de SaR y lucro. Al analizar las figuras 5.6(a), 5.6(b), 5.6(d) y 5.6(c), se puede notar fácilmente que la opción B, donde $R_{latencia}$ tiene prioridad, presenta los resultados más altos en términos de SaR logrando tasas de 93% para RS y URLLC. Adicionalmente, los servicios que no requieren estrictos requisitos de latencia como eMBB y mMTC alcanzan hasta 93% y 99%, respectivamente. Por otro lado, la figura 5.6(e) evidencia una disminución de hasta un 10% en el lucro cuando se reduce su prioridad en la recompensa total. Dado que el éxito de las RS es fundamental en el mecanismo,

(a) Variando α_1 (b) Variando α_2 (c) Variando α_3 (d) Variando α_4 (e) Lucro (Variando α_n)Figura 5.5: Variación de α

(a) Variando δ_1, δ_2 (b) Variando δ_1, δ_2 (c) Variando δ_1, δ_2 (d) Variando δ_1, δ_2 (e) Lucro (Variando δ_n)Figura 5.6: Variación δ_1, δ_2

la opción B es la más adecuada considerando que la ganancia no disminuye en gran medida y en consecuencia se obtienen valores altos de SaR .

Una vez establecidos los valores de los parámetros relacionados con el espacio de

(a) Variando capas ocultas (hl)(b) Variando neuronas (neu)(c) Variando factor de descuento (γ)(d) Variando tamaño de mini-batch (bs)(e) Variando frecuencia de actualización (τ)Figura 5.7: Variación de hl , neu , γ , bs , τ

Parámetro	Valor			
	β_1	β_2	β_3	
P_{RS}	0.25	0.5	0.25	
P_{URLLC}	0.25	0.5	0.25	
P_{eMBB}	x	0.7	0.3	
P_{mMTC}	0.25	0.5	0.25	
Recompensa de Latencia	α_1	α_2	α_3	α_4
	0.4	0.4	0.1	0.1
Recompensa Total	δ_1		δ_2	
	0.8		0.2	
Capas ocultas (hl)	2			
Neuronas (neu)	250			
Factor de descuento (γ)	0.99			
Tamaño de <i>Mini-Batch</i> (bs)	15			
Frecuencia de actualización (τ)	400			

Tabla 5.5: Configuración de Parámetros

estados y la recompensa, se configuran los hiperparámetros que componen la estructura interna del agente DRL. El conjunto de hiperparámetros considera el número de capas ocultas hls , el número de neuronas neu , el factor de descuento γ , el tamaño del *Mini-Batch* bs y la frecuencia de actualización τ . La figura 5.7(a) muestra los resultados cuando se varían el número de capas ocultas. Los resultados revelan que la recompensa total obtenida es ligeramente superior al utilizar 2 y 5 hls , mientras de la figura 5.7(b) se concluye que una capa oculta de 250 neuronas obtiene recompensas ligeramente superiores a las capas ocultas con 100, 200 y 300 neuronas.

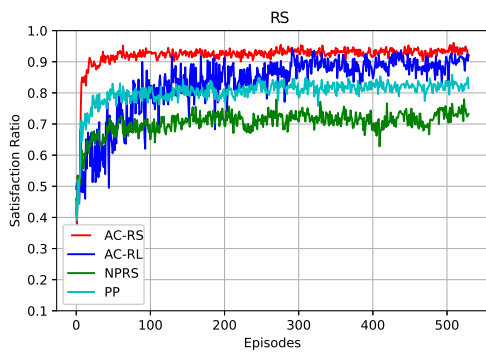
La figura 5.7(c) muestra los resultados al variar el factor de descuento γ . Para $\gamma = 0.99$, el agente DRL logra recompensas más altas que cuando γ equivale a 0.1, 0.3 y 0.5. La figura 5.7(d) revela que no hay diferencias significativas en las recompensas cuando se varía el parámetro bs . El parámetro τ indica cada cuántos pasos de tiempo se actualizan los hiperparámetros del *Target* NN. La figura 5.7(e) muestra que con $\tau = 400$ hay una ligera mejora en las recompensas en lugar de valores como 150, 300 y 500. Como resultado de la variación de todos los factores y parámetros que hacen parte del diseño del mecanismo de control de admisión, la tabla 5.5 presenta la configuración final de los valores establecidos para cada uno de ellos.

5.2 Resultados y Análisis

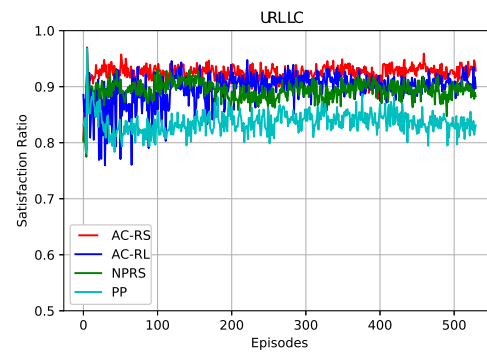
El mecanismo de AC diseñado, de ahora en adelante denominado AC-RS para este documento, fue comparado con tres propuestas: AC basado en RL (AC-RL), RS No Priorizada (NPRS) y Priorización de Lucro (PP). AC-RL implementa un agente de referencia a Q-Learning (constituyendo una línea de base con aprendizaje por refuerzo), un algoritmo de diferencia temporal tradicional de aprendizaje por refuerzo ampliamente conocido y sin modelo. NPRS es un mecanismo AC que no presenta ningún tipo de priorización para el servicio RS; es decir, todas las SLR son manejadas con total igualdad. PP es un mecanismo AC que busca incrementar la ganancia de un InP independientemente del nivel de cumplimiento de la latencia.

5.2.1 Red de 32 Nodos

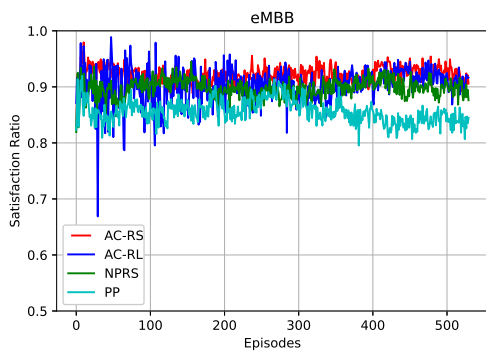
Los resultados mostrados en la figura 5.8 permiten ver una perspectiva amplia sobre el comportamiento de todos los mecanismos de AC en la topología de 32 nodos con respecto a las diferentes métricas establecidas (ver subsección 5.1.2). La figura 5.8(a) muestra que AC-RS supera las propuestas de RL, NPRS y PP en términos de SaR para el servicio RS en un 3%, 10% y 20%, respectivamente. Este comportamiento es similar en la figura 5.8(b), donde AC-RS supera en términos de SaR para el servicio URLLC a las propuestas de RL, NPRS y PP en 2%, 3% y 9%, respectivamente. Asimismo, las figuras 5.8(c) y 5.8(d) muestran que AC-RS tiene un SaR para los servicios eMBB y mMTC igual o ligeramente superior en contraste con las otras propuestas mencionadas. Estos resultados son esperados ya que todos los parámetros en AC-RS fueron ajustados para priorizar RS y URLLC, sin afectar significativamente el desempeño de eMBB y mMTC. Los resultados anteriores respaldan los valores de SaR obtenidos en la figura 5.8(e), donde el mecanismo alcanza niveles de SaR promedio (SaR promedio de todos los tipos de SLR) del 94 %, que es 2%, 6% y 7% superior a las propuestas de RL, NPRS y PP, respectivamente. Tenga en cuenta que este mecanismo muestra resultados prometedores en comparación con la literatura [19], dado que el porcentaje de satisfacción de los requisitos de QoS presentados para el caso de uso URLLC es hasta un 10% mayor.



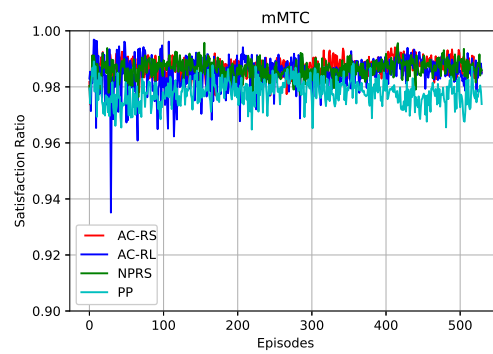
(a) SaR RS



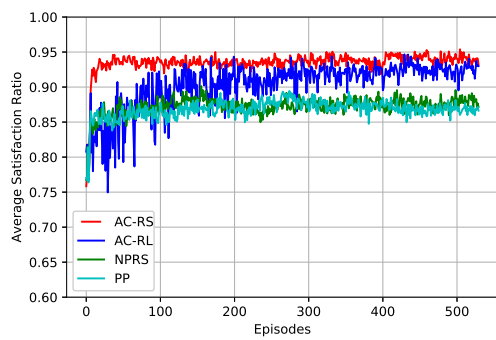
(b) SaR URLLC



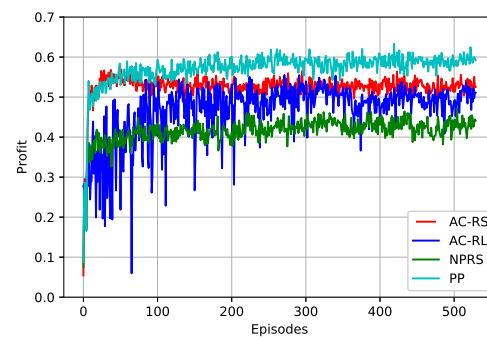
(c) SaR eMBB



(d) SaR mMTC



(e) SaR Promedio



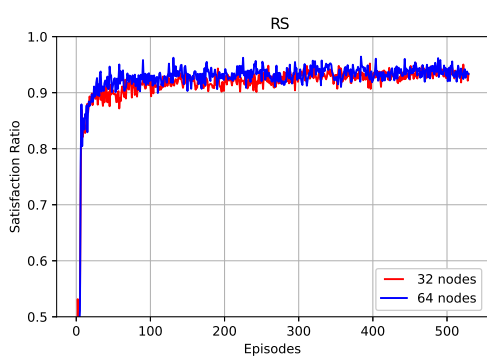
(f) Lucro

Figura 5.8: Análisis de rendimiento en topología de 32 nodos

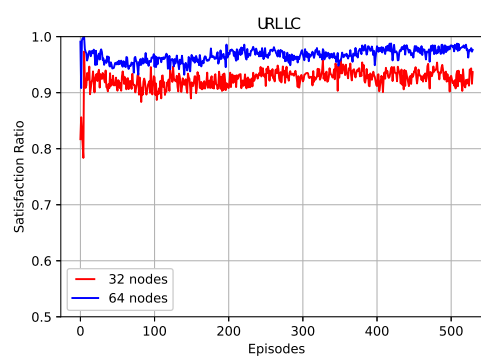
La figura 5.8(f) muestra que el AC-RS obtiene valores de lucro mayores a AC-RL en un 3%, ya que esta propuesta utiliza el concepto de redes neuronales como función de aproximación, permitiendo una generalización más granular de los estados y una capacidad predictiva más fina. En la misma figura, también se observa que la propuesta NPRS obtiene un 10% de menor de lucro, ya que su política no prioriza las SLR del servicio de cirugía remota, las cuales ofrecen un beneficio monetario superior al resto de servicios. Por otro lado, PP consigue un mejor rendimiento que AC-RS en términos de lucro, alcanzando valores de 60%. Esto se debe a que PP busca el mayor lucro posible para un InP; sin embargo, esto limita su capacidad en términos de *SaR*, alcanzando un *SaR* Promedio del 86%, dado que se reducen los niveles de satisfacción de los requisitos de QoS.

5.2.2 Red de 64 Nodos

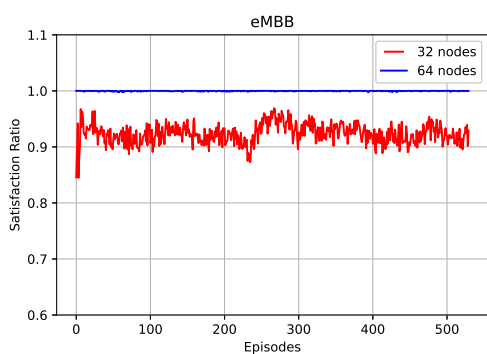
Los resultados muestran que AC-RS presenta un mejor comportamiento en la topología de 64 nodos en términos de lucro, *SaR* y *SaR* promedio (ver figura 5.9). Las figuras 5.9(a), 5.9(b), 5.9(c) y 5.9(d) muestran que los *SaR* obtenidos para URLLC, eMBB y mMTC en una topología de 64 nodos convergen a valores de 97%, 100% y 100%, respectivamente. Esto se debe a que una topología más grande tiene una gran cantidad de recursos que permiten la construcción de un mayor número de *slices*. Por lo tanto, el cumplimiento de los requisitos de QoS de una SLR aumenta proporcionalmente al número de nodos de la red. Sin embargo, es necesario resaltar que el servicio RS se mantuvo constante cuando la topología aumentó su tamaño de 32 nodos a 64 nodos, dado que, aunque la topología de 64 nodos tiene suficientes recursos de CPU y ancho de banda, los valores de latencia de los enlaces no disminuyen, por lo tanto, la posibilidad de cumplir con el requisito de latencia de los *slices* RS no aumenta. Los resultados descritos anteriormente soportan los resultados obtenidos en la figura 5.9(e), donde AC-RS en la topología de 64 nodos alcanza un *SaR* promedio del 97%. Finalmente, la figura 5.9(f) muestra que AC-RS en la topología de 64 nodos alcanza el 64% en términos de lucro de la red, esto es consistente ya que una topología de 64 nodos tiene más recursos de red para ofrecer.



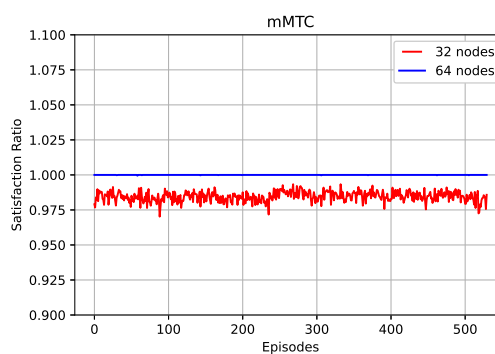
(a) SaR RS



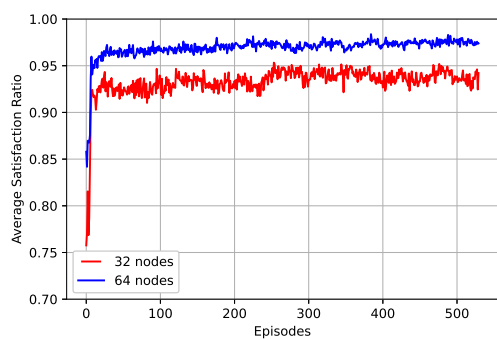
(b) SaR URLLC



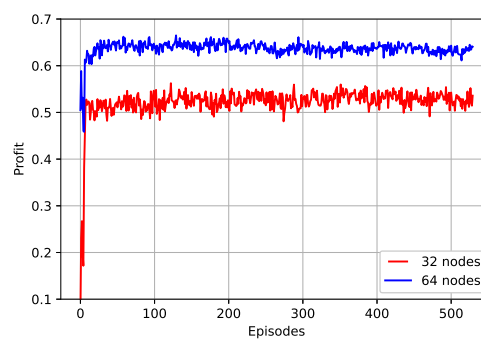
(c) SaR eMBB



(d) SaR mMTC



(e) SaR Promedio



(f) Lucro

Figura 5.9: Rendimiento del control de admisión según la topología

Capítulo 6

Conclusiones y Trabajos Futuros

6.1 Conclusiones

En este trabajo se presenta la respuesta a la pregunta: **¿Cómo realizar un control de admisión de *slices* de cirugías remotas en el núcleo de una red 5G para optimizar la latencia y el lucro de los InP?**

Para responder dicha pregunta fue necesario diseñar un Mecanismo de Control de Admisión de *slices* de red en el núcleo de una red 5G, el cual fue ajustado para priorizar el servicio de cirugía remota. Además, se implementó un prototipo *software* mediante el lenguaje de programación Python 3. Una vez implementado, se procedió a realizar el análisis de rendimiento en un ambiente simulado, considerando las métricas *SaR*, *SaR* promedio y lucro. Los resultados mostraron lo siguiente:

- Al establecer la configuración de parámetros más conveniente, RS y URLLC alcanzan un *SaR* del 93%, siendo así el más alto *SaR* alcanzado por un mecanismo AC de *slices* en el núcleo de una red 5G enfocado al servicio de cirugía remota.
- AC-RS alcanzó valores de *SaR* que superan a las propuestas AC-RL, NPSR y PP hasta en un 20%, 9%, 7% y 1,5% para los servicios RS, URLLC, eMBB

y mMTC, respectivamente. Adicionalmente, AC-RS presenta niveles de *SaR* promedio del 94%, lo cual es 2%, 6% y 7% superior a las propuestas de AC-RL, NPSR y PP.

- AC-RS logró optimizar el lucro del Proveedor de Infraestructura en un 53% y 64% para las topología de 32 y 64 nodos, respectivamente. Superando hasta en un 3% y 10% a las propuestas AC-RL y NPSR, respectivamente. Cabe resaltar que AC-RS presenta menor lucro en contraste con PP, dado que esta ultima propuesta enfoca todos sus esfuerzos en la optimización del lucro.

En general, AC-RS explota de manera mas eficiente los recursos del núcleo de una infraestructura 5G, esclareciendo la hipótesis planteada para este trabajo de grado: **Un mecanismo de control de admisión de *slices* basado en aprendizaje por refuerzo profundo podría optimizar el lucro de un InP bajo estrictos requisitos de latencia para el servicio de cirugía remota.** AC-RS presentó un mejor comportamiento en términos de *SaR* y lucro con respecto a las propuestas de referencia, dado que el parámetro latencia fue optimizado para alcanzar la mayor cantidad de *slices* de red satisfechos. Paralelamente, AC-RS logró optimizar el lucro del proveedor de infraestructura 5G, construyendo políticas de admisión que incrementan sus ingresos monetarios.

6.2 Trabajos futuros

- Implementar un mecanismo de AC de *slices* de red 5G enfocado en cirugía remota, donde se contemplen parámetros adicionales relacionados con este servicios, tales como: disponibilidad y confiabilidad.
- Implementar un mecanismo de AC de *slices* de red 5G enfocado en cirugía remota, donde se contemplen recursos del núcleo de la red y la RAN.
- Implementar un mecanismo de AC de *slices* de red 5G enfocado en cirugía remota en una herramienta de emulación *Open Source*

Bibliografía

- [1] OMS, “Fortalecimiento de la atención de emergencia, la atención quirúrgica esencial y la anestesia como componente de la cobertura sanitaria universal: Informe de la secretaría,” <https://apps.who.int/iris/handle/10665/176812>, May 2014.
- [2] S. Kumar, “Telesurgery: An audit,” in *Telesurgery*. Berlin, Heidelberg: Springer Berlin Heidelberg, january 2008, pp. 179–184.
- [3] 5G-ppp, “5g empowering vertical industries,” https://5g-ppp.eu/wp-content/uploads/2016/02/BROCHURE_5PPP_BAT2_PL.pdf, February 2016.
- [4] J. Ordóñez, P. Ameigeiras, D. Lopez, J. Ramos, J. Lorca, and J. Folgueira, “Network slicing for 5g with sdn/nfv: Concepts, architectures and challenges,” *IEEE Communications Magazine*, vol. 55, 03 2017.
- [5] M. A. Siddiqi, H. Yu, and J. Joung, “5g ultra-reliable low-latency communication implementation challenges and operational issues with iot devices,” *Electronics*, vol. 8, no. 9, 2019. [Online]. Available: <https://www.mdpi.com/2079-9292/8/9/981>
- [6] Cisco, {<https://www.cisco.com/>}.
- [7] ITU-T, “The tactile internet,” https://www.itu.int/dms_pub/itu-t/opb/gen/T-GEN-TWATCH-2014-1-PDF-E.pdf, 2014.
- [8] Q. Zhang, J. Liu, and G. Zhao, “Towards 5g enabled tactile robotic telesurgery,” *arXiv: Networking and Internet Architecture*, vol. abs/1803.03586, 2018.

-
- [9] A. Ahad, M. Tahir, and K. A. Yau, “5g-based smart healthcare network: Architecture, taxonomy, challenges and future research directions,” *IEEE Access*, vol. 7, pp. 100 747–100 762, 2019.
- [10] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A survey on low latency towards 5g: Ran, core network and caching solutions,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3098–3130, 2018.
- [11] Nokia, “Low latency in 4.9g/5g,” <https://gsacom.com/paper/low-latency-4-9g5g-solutions-millisecond-latency/>, 2017.
- [12] ITU-T, “Series y: Global information infrastructure, internet protocol aspects,next-generation networks, internet of things and smart cities,” *Recommendation ITU-T Y.3100*, september 2017.
- [13] V. Sciancalepore, X. Costa-Perez, and A. Banchs, “Rl-nsb: Reinforcement learning-based 5g network slice broker,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1543–1557, 2019.
- [14] S. Meng, Z. Wang, H. Ding, S. Wu, X. Li, P. Zhao, C. Zhu, and X. Wang, “Ran slice strategy based on deep reinforcement learning for smart grid,” in *2019 Computing, Communications and IoT Applications (ComComAp)*, 2019, pp. 6–11.
- [15] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno, “Flexible resource block allocation to multiple slices for radio access network slicing using deep reinforcement learning,” *IEEE Access*, vol. 8, pp. 68 183–68 198, 2020.
- [16] R. Li, Z. Zhao, Q. Sun, C. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, “Deep reinforcement learning for resource management in network slicing,” *IEEE Access*, vol. 6, pp. 74 429–74 441, 2018.
- [17] Y. Abiko, D. Mochizuki, T. Saito, D. Ikeda, T. Mizuno, and H. Mineno, “Proposal of allocating radio resources to multiple slices in 5g using deep reinforcement learning,” in *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, 2019, pp. 1–2.

-
- [18] M. Jiang, M. Condoluci, and T. Mahmoodi, “Network slicing management prioritization in 5g mobile systems,” in *European Wireless 2016; 22th European Wireless Conference*, Oulu, Finland, Finland, May 2016, pp. 1–6.
- [19] A. Kammoun, N. Tabbane, G. Diaz, and N. Achir, “Admission control algorithm for network slicing management in sdn-nfv environment,” *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 1–6, May 2018.
- [20] K. Noroozi, M. Karimzadeh-Farshbafan, and V. Shah-Mansouri, “Service admission control for 5g mobile networks with ran and core slicing,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, USA, 2019, pp. 1–6.
- [21] B. Han, V. Sciancalepore, D. Feng, X. Costa-Perez, and H. D. Schotten, “A utility-driven multi-queue admission control solution for network slicing,” in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Paris, France, France, April 2019, pp. 55–63.
- [22] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, “Network slicing: Recent advances, taxonomy, requirements, and open research challenges,” *IEEE Access*, vol. 8, pp. 36 009–36 028, 2020.
- [23] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, “Network slicing in 5g: Survey and challenges,” *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, May 2017.
- [24] 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.
- [25] A. Ksentini and N. Nikaein, “Toward enforcing network slicing on ran: Flexibility and resources abstraction,” *IEEE Communications Magazine*, vol. 55, no. 6, pp. 102–108, 2017.

-
- [26] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, and D. I. Kim, “Applications of deep reinforcement learning in communications and networking: A survey,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.
- [27] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, “Optimising 5g infrastructure markets: The business of network slicing,” in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, Atlanta, GA, EE. UU., May 2017, pp. 1–9.
- [28] B. Han, D. Feng, and H. D. Schotten, “A markov model of slice admission control,” *IEEE Networking Letters*, vol. 1, no. 1, pp. 2–5, 2019.
- [29] H. Chen, X. Li, Y. Zheng, and L. Lv, “A robust multi-pedestrians tracker based on the improved mdp strategy,” in *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2018, pp. 350–354.
- [30] 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.
- [31] M. R. Raza, C. Natalino, P. Öhlen, L. Wosinska, and P. Monti, “Reinforcement learning for slicing in a 5g flexible ran,” *Journal of Lightwave Technology*, vol. 37, no. 20, pp. 5161–5169, 2019.
- [32] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, and X. Costa-Pérez, “A machine learning approach to 5g infrastructure market optimization,” *IEEE Transactions on Mobile Computing*, vol. 19, no. 3, pp. 498–512, March 2020.
- [33] B. Khodapanah, A. Awada, I. Viering, A. N. Barreto, M. Simsek, and G. Fettweis, “Slice management in radio access network via deep reinforcement learning.”
- [34] M. Iwamura, “Ngmn view on 5g architecture,” in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1–5.

-
- [35] A. Al-Dulaimi, X. Wang, and C. I, “Introduction,” in *5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management*, 2018, pp. 1–12.
- [36] METIS, “Updated scenarios requirements and kpls for 5g mobile and wireless system with recommendations for future investigations,” April 2015.
- [37] S. E. Elayoubi, M. Fallgren, P. Spapis, G. Zimmermann, D. Martín-Sacristán, C. Yang, S. Jeux, P. Agyapong, L. Campoy, Y. Qi, and S. Singh, “5g service requirements and operational use cases: Analysis and metis ii vision,” in *2016 European Conference on Networks and Communications (EuCNC)*, 2016, pp. 158–162.
- [38] A. Al-Dulaimi, X. Wang, and C. I, “Emerging technologies in software, hardware, and management aspects toward the 5g era: Trends and challenges,” in *5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management*, 2018, pp. 13–50.
- [39] W. Lee, T. Na, and J. Kim, “How to create a network slice? - a 5g core network perspective,” in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, 2019, pp. 616–619.
- [40] J. F. Kurose and K. W. Ross, *Computer Networking A Top-Down Approach*. England: Pearson Global Edition, 2017.
- [41] ONF, “Sdn architecture overview,” November 2014.
- [42] E. I. S. G. (ISG-NFV), “Network functions virtualisation (nfv); terminology for main concepts in nfv,” july 2020.
- [43] C. Cobo, “El concepto de tecnologías de la información. benchmarking sobre las definiciones de las tic en la sociedad del conocimiento,” *Zer: Revista de estudios de comunicación = Komunikazio ikasketen aldizkaria*, vol. 14, pp. 295–318, 01 2009.
- [44] E. I. S. G. I. NFV, “Network functions virtualisation; an introduction, benefits, enablers, challenges call for action,” october 2012.

- [45] J. Sachs, L. A. A. Andersson, J. Araújo, C. Curescu, J. Lundsjö, G. Rune, E. Steinbach, and G. Wikström, “Adaptive 5g low-latency communication for tactile internet services,” *Proceedings of the IEEE*, vol. 107, no. 2, pp. 325–349, 2019.
- [46] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, “Tactile-internet-based telesurgery system for healthcare 4.0: An architecture, research challenges, and future directions,” *IEEE Network*, vol. 33, no. 6, pp. 22–29, 2019.
- [47] H. Shariatmadari, Z. Li, K. Hugl, M. A. Uusitalo, and R. Jantti, “Statistical analysis of downlink transmissions for ultra-reliable low-latency communications,” in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 2018, pp. 1–5.
- [48] P. Choi, R. Oskouian, and R. Tubbs, “Telesurgery: Past, present, and future,” *Cureus*, vol. 10, 05 2018.
- [49] S. Georgoulas, P. Trimintzios, and G. Pavlou, “Admission control placement in differentiated services networks,” in *Proceedings. ISCC 2004. Ninth International Symposium on Computers And Communications (IEEE Cat. No.04TH8769)*, vol. 2, 2004, pp. 816–821 Vol.2.
- [50] R. Yavatkar, D. E. Pendarakis, and R. Guérin, “A framework for policy-based admission control,” *RFC*, vol. 2753, pp. 1–20, 2000.
- [51] J. Majín, R. Muñoz, and O. Calderón, “Impacto del control de admisión en el desempeño de una red lte,” *Puente revista científica universidad pontificia bolivariana seccional bucaramanga*, vol. 9, no. 1, 04 2015.
- [52] E. Mykoniati, C. Charalampous, P. Georgatsos, T. Damilatis, D. Goderis, P. Trimintzios, G. Pavlou, and D. Griffin, “Admission control for providing qos in diffserv ip networks: the tequila approach,” *IEEE Communications Magazine*, vol. 41, no. 1, pp. 38–44, 2003.
- [53] A. Kaloxylou, A. Gavras, and R. De Peppe, “Empowering Vertical Industries through 5G Networks - Current Status and Future Trends,” Aug. 2020. [Online]. Available: <https://doi.org/10.5281/zenodo.3698113>

-
- [54] E. Chromy, M. Jadron, and T. Behul, “Admission control methods in ip networks,” *Advances in Multimedia*, vol. 2013, 03 2013.
- [55] H. Al-Sahaf, Y. Bi, Q. Chen, A. Lensen, Y. Mei, Y. Sun, B. Tran, B. Xue, and M. Zhang, “A survey on evolutionary machine learning,” *Journal of the Royal Society of New Zealand*, vol. 49, no. 2, pp. 205–228, 2019.
- [56] D. Rafique and L. Velasco, “Machine learning for network automation: overview, architecture, and applications [invited tutorial],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D126–D143, 2018.
- [57] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.
- [58] S. Ayoubi, N. Limam, M. A. Salahuddin, N. Shahriar, R. Boutaba, F. Estrada-Solano, and O. M. Caicedo, “Machine learning for cognitive network management,” *IEEE Communications Magazine*, vol. 56, no. 1, pp. 158–165, 2018.
- [59] 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.
- [60] 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.
- [61] 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.
- [62] G. Karatas, O. Demir, and O. Koray Sahingoz, “Deep learning in intrusion detection systems,” in *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, 2018, pp. 113–116.

- [63] P. Sherkhane and D. Vora, “Survey of deep learning software tools,” in *2017 International Conference on Data Management, Analytics and Innovation (ICDMAI)*, 2017, pp. 236–238.
- [64] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis : A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 01 2018.
- [65] L. Kuang, L. Yang, and Y. Liao, “An integration framework on cloud for cyber physical social systems big data,” *IEEE Transactions on Cloud Computing*, pp. 1–1, 2015.
- [66] B. Mao, Z. M. Fadlullah, F. Tang, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, “Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning,” *IEEE Transactions on Computers*, vol. 66, no. 11, pp. 1946–1960, 2017.
- [67] F. Toqué, E. Côme, M. K. El Mahrssi, and L. Oukhellou, “Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1071–1076.
- [68] C. Siripanpornchana, S. Panichpapiboon, and P. Chaovalit, “Travel-time prediction with deep learning,” in *2016 IEEE Region 10 Conference (TENCON)*, 2016, pp. 1859–1862.
- [69] J. Rogers, D. Simmons, M. Shah, C. Rowland, and Y. Shang, “Deep learning at your fingertips,” in *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2019, pp. 1–4.
- [70] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–33, 02 2015.

- [71] S. Thrun and A. Schwartz, “Issues in using function approximation for reinforcement learning,” in *Proceedings of the 1993 Connectionist Models Summer School*, D. T. J. E. M. Mozer, P. Smolensky and A. Weigend, Eds. Erlbaum Associates, January 1993.
- [72] I. A. T. Schaul, J. Quan and D. Silver, “Prioritized experience replay,” vol. 4, pp. 1–21, 02 2016. [Online]. Available: <http://arxiv.org/abs/1511.05952>
- [73] Z. Wang, N. Freitas, and M. Lanctot, “Dueling network architectures for deep reinforcement learning,” 11 2015.
- [74] V. Mnih, A. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” 02 2016.
- [75] M. Bellemare, W. Dabney, and R. Munos, “A distributional perspective on reinforcement learning,” 07 2017.
- [76] M. Fortunato, M. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin, C. Blundell, and S. Legg, “Noisy networks for exploration,” 06 2017.
- [77] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver, “Rainbow: Combining improvements in deep reinforcement learning,” *ArXiv*, vol. abs/1710.02298, 2018.
- [78] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” *31st International Conference on Machine Learning, ICML 2014*, vol. 1, 06 2014.
- [79] M. Hausknecht and P. Stone, “Deep recurrent q-learning for partially observable mdps,” 07 2015.
- [80] D. Zhao, Haitao Wang, Kun Shao, and Y. Zhu, “Deep reinforcement learning with experience replay based on sarsa,” in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–6.

-
- [81] W. Wang, J. Hao, Y. Wang, and M. Taylor, “Towards cooperation in sequential prisoner’s dilemmas: a deep multiagent reinforcement learning approach,” *CoRR*, vol. abs/1803.00162, 2018. [Online]. Available: <http://arxiv.org/abs/1803.00162>
- [82] 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.
- [83] G. Stampa, M. Arias, D. Sanchez-Charles, V. Muntés-Mulero, and A. Cabellos, “A deep-reinforcement learning approach for software-defined networking routing optimization,” *CoRR*, vol. abs/1709.07080, 2017. [Online]. Available: <http://arxiv.org/abs/1709.07080>
- [84] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, “Experience-driven networking: A deep reinforcement learning based approach,” *CoRR*, vol. abs/1801.05757, 2018. [Online]. Available: <http://arxiv.org/abs/1801.05757>
- [85] K. Yokoyama and K. Morioka, “Autonomous mobile robot with simple navigation system based on deep reinforcement learning and a monocular camera,” in *2020 IEEE/SICE International Symposium on System Integration (SII)*, 2020, pp. 525–530.
- [86] X. Ruan, D. Ren, X. Zhu, and J. Huang, “Mobile robot navigation based on deep reinforcement learning,” in *2019 Chinese Control And Decision Conference (CCDC)*, 2019, pp. 6174–6178.
- [87] R. Tan, J. Zhou, H. Du, S. Shang, and L. Dai, “An modeling processing method for video games based on deep reinforcement learning,” in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, 2019, pp. 939–942.
- [88] 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.

- [89] I. Kamaldinov and I. Makarov, “Deep reinforcement learning in match-3 game,” in *2019 IEEE Conference on Games (CoG)*, 2019, pp. 1–4.
- [90] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, “Resource management with deep reinforcement learning,” in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, 2016, pp. 50–56.
- [91] I. K. Gupta, “A hybrid ga-gsa algorithm to solve multidimensional knapsack problem,” in *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, 2018, pp. 1–6.
- [92] R. S. Sutton, D. Precup, and S. Singh, “Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning,” *Artificial Intelligence*, vol. 112, no. 1, pp. 181 – 211, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0004370299000521>
- [93] J.-X. Peng, K. Li, and G. W. Irwin, “A new jacobian matrix for optimal learning of single-layer neural networks,” *IEEE Transactions on Neural Networks*, vol. 19, no. 1, pp. 119–129, 2008.
- [94] M. R. Raza, A. Rostami, L. Wosinska, and P. Monti, “A slice admission policy based on big data analytics for multi-tenant 5g networks,” *Journal of Lightwave Technology*, vol. 37, no. 7, pp. 1690–1697, 2019.
- [95] V. Perez and M. Santís-Chaves, “Interfaces hÁpticas: Sistemas cinestÉsicos vs. sistemas tÁctiles,” *Revista EIA*, vol. 13, 03 2017.
- [96] N. Etsi, “Etsi gs nfv 002 v1. 1.1 network functions virtualization (nfv),” *Architecture and Framework: ONF*, 2013.
- [97] W. Jácome, O. Caicedo Rendon, and N. Fonseca, “Admission control for 5g network slicing based on (deep) reinforcement learning,” 04 2021.
- [98] G. Brown, “Service-based architecture for 5g core networks,” *A Heavy Reading white paper produced for Huawei Technologies Co. Ltd. Online: <https://www.huawei.com/en/press-events/news/2017/11/HeavyReading-WhitePaper-5G-Core-Network>*, vol. 1, pp. 1–12, 2017.

-
- [99] R. W. Wolff, “Poisson arrivals see time averages,” *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982. [Online]. Available: <http://www.jstor.org/stable/170165>
- [100] D. M. Casas-Velasco, O. M. C. Rendon, and N. L. S. da Fonseca, “Intelligent routing based on reinforcement learning for software-defined networking,” *IEEE Transactions on Network and Service Management*, pp. 1–1, 2020.
- [101] M. Sewak, *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*, 1st ed. Springer Publishing Company, Incorporated, 2019.
- [102] Z. Wang and A. C. Bovik, “Mean squared error: Love it or leave it? a new look at signal fidelity measures,” *IEEE Signal Processing Magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [103] Y. Sai, R. Jinxia, and L. Zhongxia, “Learning of neural networks based on weighted mean squares error function,” in *2009 Second International Symposium on Computational Intelligence and Design*, vol. 1, 2009, pp. 241–244.
- [104] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” 09 2015.
- [105] J. Zhang, “Gradient descent based optimization algorithms for deep learning models training,” 2019.
- [106] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–33, 02 2015.
- [107] S. R. White, J. E. Hanson, I. Whalley, D. M. Chess, and J. O. Kephart, “An architectural approach to autonomic computing,” in *International Conference on Autonomic Computing, 2004. Proceedings.*, 2004, pp. 2–9.
- [108] M. Erel-Özçevik and B. Canberk, “Road to 5g reduced-latency: A software defined handover model for embb services,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8133–8144, 2019.

-
- [109] qualcomm, “Ultra-reliable low-latency 5g for industrial automation,” <https://www.qualcomm.com/media/documents/files/read-the-white-paper-by-heavy-reading.pdf/>, 2019.
- [110] S. R. Pokhrel, J. Ding, J. Park, O. S. Park, and J. Choi, “Towards enabling critical mmTC: A review of URLLC within mmTC,” *IEEE Access*, vol. 8, pp. 131 796–131 813, 2020.
- [111] R. Albert and A.-L. Barabási, “Topology of evolving networks: local events and universality,” *Physical rev. letters*, vol. 85, no. 24, p. 5234, 2000.
- [112] A. Hagberg, P. Swart, and D. S Chult, “Exploring network structure, dynamics, and function using networkx,” Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

**Control de admisión basado en aprendizaje por
refuerzo profundo para el servicio de cirugía
remota en 5G**



ANEXOS

Trabajo de grado

Santiago de Jesús Martínez Semanate

Jose Luis Rincón Zapata

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, 2021*

ANEXO A

El Anexo A presenta el enlace que redirecciona a el repositorio de GIT HUB en el cuál está alojado el código de fuente de nuestro Control de admision basado en aprendizaje por refuerzo profundo para el servicio de cirugía remota en 5G.

<https://github.com/JoseLuis9710/tesis>

ANEXO B

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

- **Santiago de Jesus Martinez Semanate, Jose Luis Rincón Zapata, Oscar Mauricio Caicedo Rendon and Johanna Andrea Hurtado Sanchez. Admission Control based on Deep Reinforcement Learning for Remote Surgery in 5G. IEEE Latin America Transactions 2021.**
 - Estado: Enviado.
 - Clasificación: A2
 - Factor de Impacto : 0.729
 - Índice H: 26 Scimago