ISSN 1688-2806

Universidad de la República
Facultad de Ingeniería

Aprendizaje Profundo por Refuerzo Aplicado al Control de Acceso en Redes IEEE 802.11

Tesis presentada a la Facultad de Ingeniería de la Universidad de la República por

Fabián Frommel

en cumplimiento parcial de los requerimientos para la obtención del título de Magister en Ingeniería Eléctrica.

Directores de Tesis
Dr. Ing. Germán Capdehourat ..... Universidad de la República
Dr. Ing. Federico La Rocca ..... Universidad de la República

Tribunal
Dr. Ing. Juan Bazerque ..... Universidad de la República
Dr. Ing. Claudina Rattaro ..... Universidad de la República
Dr. Ing. Matías Richart ..... Universidad de la República

Director Académico
Dr. Ing. Germán Capdehourat ..... Universidad de la República

Montevideo
jueves 30 de junio, 2022

Aprendizaje Profundo por Refuerzo Aplicado al Control de Acceso en Redes IEEE 802.11, Fabián Frommel.

ISSN 1688-2806

Esta tesis fue preparada en LaTeX usando la clase iietesis (v1.1).
Contiene un total de 96 páginas.
Compilada el martes 12 julio, 2022.
http://iie.fing.edu.uy/

Agradecimientos

En primer lugar, quiero agradecer a mis directores de tesis, por todo el tiempo dedicado y la orientación brindada, necesaria para el desarrollo y culminación de este trabajo de investigación.

A la Universidad de la República por brindarme la posibilidad de continuar desarrollándome en el ámbito académico y personal.

A Ceibal (Centro Ceibal para el Apoyo a la Educación de la Niñez y la Adolescencia) por el apoyo recibido durante todo este proceso y, en particular, por el préstamo del lugar físico, los equipos y otros insumos para las evaluaciones con equipos comerciales.

Por último, quiero dar un agradecimiento especial a mi pareja, familia y amigos por el acompañamiento incondicional ofrecido a lo largo de la realización de mi Maestría.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

En estos últimos tiempos, las tecnologías de la información y la comunicación, apalancadas por la masificación en el acceso a internet, han modificado el comercio, la educación, el gobierno, la salud e incluso la forma en que las personas se relacionan afectivamente. Esta evolución se vio acelerada todavía más a raíz de la pandemia de COVID-19, donde muchas de las actividades tuvieron que ser migradas a una modalidad virtual (e.g. teletrabajo, videoconferencias, clases remotas). Las redes WLAN (Wireless Local Area Network) son probablemente la tecnología de acceso a internet más utilizada a nivel mundial y, en particular, el estándar IEEE 802.11, comercialmente conocido como Wi-Fi, ha sido el que ha proliferado como estándar de base para estas redes. Desde su primera publicación, se ha ido actualizando con el objetivo de adaptarse a las demandas de más dispositivos, más conexiones y mayores velocidades. Recientemente, se publicó su última enmienda, la IEEE 802.11ax, la cual introduce cambios radicales a nivel de acceso al medio en busca de una mayor eficiencia en el uso del espectro. Con ella, se plantean una serie de desafíos respecto de la asignación de recursos entre dispositivos de la nueva enmienda y de enmiendas anteriores (*legacy*), que siguen operando con el mecanismo de acceso al medio tradicional.

Por otro lado, los avances en la capacidad de procesamiento y de almacenamiento de datos de los servidores de la actualidad habilitan la aplicación de técnicas de aprendizaje automático en diversas áreas de la industria. En particular, dentro de las telecomunicaciones, se vienen utilizando ampliamente para la optimización de procesos o recursos. Dada esta realidad y los desafíos mencionados respecto del reparto de recursos entre dispositivos 802.11ax y *legacy*, en el presente trabajo se estudia el potencial de la aplicación de técnicas de aprendizaje profundo por refuerzo (DRL) a la optimización del control de acceso al medio en redes IEEE 802.11. Este enfoque presenta ventajas respecto de los enfoques tradicionales (e.g. uso de modelos analíticos), ya que estos últimos funcionan bien solo bajo ciertas suposiciones y configuraciones cuasiestáticas o están limitados en su capacidad de generalización.

En esta tesis se realizó una revisión y diagnóstico de la situación actual respecto de la nueva enmienda de IEEE 802.11, mediante simulaciones y pruebas con equipos comerciales. Las pruebas exhaustivas realizadas revelaron cierta inmadurez de las implementaciones de 802.11ax por el momento. Posteriormente, se propuso el método Enhanced - Centralized Contention Window Optimization with Deep Reinforcement Learning (E-CCOD), para aplicar DRL a la optimización del rendimiento de redes IEEE 802.11 mediante la predicción correcta de los

valores de ventana de contención (un parámetro clave del mecanismo de control de acceso de estas redes). Se constató el funcionamiento satisfactorio de este método en distintos escenarios de operación realistas: envíos UDP y TCP, sentidos uplink, downlink y bidireccional, y variaciones en la cantidad de clientes y en el tráfico cursado. Por último, se lo extendió para operar en redes donde coexisten clientes de la nueva versión del estándar y de versiones anteriores, y se lo puso a prueba en un ejemplo de aplicación. A partir de los resultados obtenidos, es posible afirmar la viabilidad de aplicar técnicas de DRL a la optimización del control de acceso en redes IEEE 802.11.

Tabla de contenidos

Agradecimientos I

Resumen III

1. Introducción 1
1.1. Contexto y Motivación 1
1.2. Trabajo Realizado y Principales Aportes 3
1.3. Organización del Documento 5

2. Evolución del Estándar IEEE 802.11 7
2.1. Repaso Histórico 7
2.2. El Estándar IEEE 802.11ax 9
2.2.1. Nuevas Funcionalidades 10
2.3. Desempeño de IEEE 802.11 15
2.3.1. Modelo Teórico Legacy 15
2.3.2. Evaluación de Desempeño de IEEE 802.11ax 17

3. Aprendizaje Profundo por Refuerzo 29
3.1. Inteligencia Artificial y Aprendizaje Automático 29
3.2. Aprendizaje Profundo 31
3.3. Aprendizaje por Refuerzo 33
3.3.1. Algoritmos 34
3.4. Aprendizaje Profundo por Refuerzo 37
3.4.1. Algoritmos 37

4. Aplicación de Aprendizaje Profundo por Refuerzo al Control de Acceso en Redes IEEE 802.11 41
4.1. Trabajo Relacionado 41
4.2. Desempeño del Agente para el Caso UDP 44

5. Presentación y Evaluación del Método E-CCOD para la Optimización del Control de Acceso en IEEE 802.11 51
5.1. Presentación del Método 51
5.2. Evaluación de Desempeño del Método 55
5.2.1. Caso UDP 55
5.2.2. Caso TCP 57

5.2.3. Caso con Variaciones en el Tráfico Cursado 61
5.3. Aplicación del Método a Redes con Clientes IEEE 802.11ax y Legacy 63

6. Conclusiones y Trabajo Futuro **69**
6.1. Conclusiones 69
6.2. Trabajo Futuro 71

Referencias **73**

Índice de tablas **81**

Índice de figuras **83**

Capítulo 1

Introducción

1.1. Contexto y Motivación

La importancia que ha adquirido internet en estos últimos tiempos es indiscutible. Se estima que en enero de 2022 la cantidad de usuarios fue de 4,95 mil millones, lo que equivale a un 62,5 % de la población mundial. Mientras tanto, el tiempo promedio diario dedicado al uso de internet fue de casi 7 horas en todos los dispositivos a nivel mundial [1]. Las tecnologías de la información y la comunicación (TIC), apalancadas por la masificación en el acceso a internet, han modificado el comercio, la educación, el gobierno, la salud e incluso la forma en que las personas se relacionan afectivamente. Además, durante el período de confinamiento provocado por la pandemia de Corona Virus Disease (COVID-19), el uso de las TIC cobró todavía más relevancia, puesto que muchas de las actividades que se desarrollaban de manera presencial, tuvieron que ser migradas a una modalidad virtual (e.g. teletrabajo, videoconferencias, clases remotas). Asimismo, permitió a las personas acercarse a quienes estaban lejos de sus hogares, familias, parientes y amigos manteniendo la distancia social.

En este contexto, la educación emerge como uno de los verticales más importantes para la incorporación de las TIC, con Ceibal como ejemplo a nivel nacional [2]. Esta organización fue creada en 2007 como un plan de inclusión e igualdad de oportunidades con el objetivo de apoyar con tecnología las políticas educativas uruguayas. Desde su implementación, cada niño, niña y adolescente que ingresa al sistema educativo público en todo el país accede a un dispositivo para su uso personal con conexión a internet desde el centro educativo. Hasta agosto de 2021 se han entregado 2.581.377 de laptops y tablets y 83.939 placas programables Micro:bit[1], y se cuenta con 3023 instituciones educativas con acceso a internet y 1.498 con servicio de videoconferencia [3]. Con respecto al uso de plataformas educativas, durante la etapa de virtualidad, la demanda por parte de la comunidad educativa se incrementó notoriamente, alcanzando el pico de 784.383 usuarios. En la actualidad, habiéndose retomado las clases presenciales, el uso de estas plataformas sigue siendo masivo y significativamente más alto que en 2019 [4].

[1] https://microbit.org/

Las redes WLAN (Wireless Local Area Network) son la tecnología de acceso a internet utilizada en todos los centros educativos de Uruguay (y probablemente la más usada a nivel mundial), ya que brinda movilidad al usuario y rapidez y flexibilidad para su instalación. En particular, el estándar 802.11 de la IEEE (Institute of Electrical and Electronics Engineers), comercialmente conocido como Wi-Fi, ha sido el que ha proliferado como estándar de base para estas redes, gracias a su bajo costo y su desempeño razonable [5]. Desde su primera publicación en 1997, se han liberado múltiples enmiendas (e.g. a, b, g, e, n, ac, ax) con el objetivo de adaptarse a las demandas de más dispositivos, más conexiones y mayores velocidades. Por ejemplo, las primeras enmiendas publicadas en los años 2000 ofrecían velocidades de tráfico máximas del orden de los 10 Mbps; mientras que las últimas, publicadas dos décadas más tarde, permiten alcanzar velocidades que superan 1 Gbps (se ha logrado aumentar en 2 órdenes de magnitud las velocidades ofrecidas).

El pasado 9 de febrero de 2021 se aprobó oficialmente la última enmienda [6], la IEEE 802.11ax, la cual introduce cambios significativos para mejorar la eficiencia en el uso del espectro y el rendimiento por área cubierta, con foco en escenarios densos y heterogéneos. Para el diseño de este estándar se consideró cierto aprendizaje de los errores cometidos en el pasado, donde varias de las mejoras introducidas en enmiendas anteriores no tenían luego un impacto significativo en despliegues reales, dadas las condiciones exigentes que se requerían (que solo eran alcanzables en ambientes de laboratorio). Uno de los principales cambios de 802.11ax es en el mecanismo de acceso al medio, el cual se modifica para que la infraestructura de red (i.e. los puntos de acceso -APs-) sea la encargada de establecer cuándo y con qué recursos los clientes pueden transmitir sus tramas, de manera análoga a lo que ocurre en LTE (Long Term Evolution) [7].

Si bien en esta nueva enmienda de IEEE 802.11 se modifica radicalmente el mecanismo de acceso al medio, el mecanismo básico, el CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), que ha sido utilizado desde las primeras enmiendas, se mantiene. Esto es debido a que la retrocompatibilidad siempre ha sido uno de los objetivos del estándar, lo que da lugar a una etapa de transición donde dispositivos de las versiones más nuevas conviven sin inconvenientes con otros de versiones anteriores (*legacy*). Como se verá en los siguientes capítulos, CSMA/CA es un mecanismo robusto frente a cambios en la red y de baja complejidad computacional; no obstante, puede conducir a una operación alejada de la óptima, sobre todo en escenarios de alta densidad, que cada vez son más comunes en la realidad actual [8]. En suma, el nuevo acceso al medio y la etapa de transición donde coexistan dispositivos 802.11ax y *legacy* (los cuales no implementan el nuevo acceso al medio) plantean una serie de desafíos adicionales respecto de una asignación de recursos óptima y justa, los cuales no son abordados en el nuevo estándar.

Por otro lado, la evolución que se ha dado en internet y en las tecnologías de acceso ha sido acompañada y potenciada por los avances a nivel de hardware; en particular, por las mejoras sustanciales en la capacidad de procesamiento y de almacenamiento de datos de los servidores (físicos y en la nube). Gracias a esto, la aplicación de técnicas de aprendizaje automático (ML) en diversas áreas de la

industria está en constante crecimiento y expansión. Ejemplos de este fenómeno son el descubrimiento y fabricación de fármacos y la generación de tratamientos personalizados en la medicina, la detección de fraude y la orientación hacia los clientes más importantes en las finanzas, la recomendación de productos y la atención al cliente mejorada en las ventas minoristas (*retail*), la variación dinámica de precios y el análisis de los sentimientos en el turismo y múltiples aplicaciones en redes sociales para atraer cada vez a más usuarios y retenerlos, entre otros [9]. Por su parte, dentro de las telecomunicaciones, el ML ha mostrado un gran potencial para asistir en la detección de anomalías, mantenimiento predictivo, clasificación de reportes de fallas, garantía de acuerdos de nivel de servicio y optimización de redes [10].

Dado el creciente aumento del uso de técnicas de ML para resolver problemáticas de la industria, y en particular de las telecomunicaciones, y los desafíos mencionados respecto de la asignación de recursos que trae consigo la publicación de IEEE 802.11ax, en el presente trabajo se estudia el potencial de la aplicación de aprendizaje profundo por refuerzo (DRL) a la optimización del control de acceso al medio en redes IEEE 802.11. DRL es un área de ML que consiste en aprender mediante la interacción (por prueba y error) con el entorno y resulta adecuado para abordar problemas de este estilo [11].

1.2. Trabajo Realizado y Principales Aportes

El foco principal de este trabajo de investigación es estudiar el potencial de la aplicación de DRL a la optimización del control de acceso al medio en redes IEEE 802.11. Como se mencionó, con la reciente publicación de su última enmienda (la IEEE 802.11ax), que introduce cambios radicales a nivel de acceso al medio, se plantean nuevos desafíos respecto de la asignación de recursos entre dispositivos nuevos y *legacy*, que se suman a los desafíos ya existentes en el mecanismo de control de acceso tradicional (operación alejada de la óptima en escenarios de alta densidad).

Previo al trabajo con DRL, se realizó una revisión y diagnóstico de la situación actual respecto de la nueva enmienda de IEEE 802.11. Para ello se llevó a cabo una revisión bibliográfica de la misma para comprender en profundidad sus objetivos y las principales funcionalidades que se introducen o se extienden de enmiendas anteriores. Además, se realizó una evaluación de desempeño mediante simulaciones y pruebas con equipos comerciales, con foco en escenarios educativos (i.e. salón de clases). Este tipo de escenarios, además de ser de especial interés para Ceibal, se destaca por su alta densidad, con una gran cantidad de dispositivos conectados en un entorno reducido como lo es un aula, donde no solo hay laptops y tablets, sino que también nuevos elementos que se siguen incorporando como tecnología educativa (pantallas interactivas, robots, placas programables, sensores, etc.). Las pruebas exhaustivas realizadas revelaron una inmadurez tanto de las versiones actuales de los simuladores de red como de las implementaciones de 802.11ax en los equipos comerciales por el momento; sin embargo, es de esperar que estos problemas se resuelvan con las sucesivas actualizaciones de los simuladores y del

software y hardware de equipamiento de red y de dispositivos terminales.

Más allá de estos resultados preliminares y que, con el paso del tiempo y con la evolución de los equipos, el desempeño del nuevo estándar mejore, este último no especifica cómo será el reparto de recursos entre dispositivos de la nueva versión y *legacy*. Esto puede conducir a un fenómeno análogo al conocido como *anomalía de 802.11*, donde CSMA/CA no proporciona equidad de tiempo de aire para todas las estaciones (STAs), ya que, debido a su diseño, el desempeño máximo alcanzable por cualquier STA de la red está limitado por la STA más lenta [12]. Por el contrario, podría ocurrir que los dispositivos de versiones anteriores queden excluidos injustamente del uso del canal. Para contribuir al estudio de esta temática, se propone el método Enhanced - Centralized Contention Window Optimization with Deep Reinforcement Learning (E-CCOD), basado en DRL, para la optimización del mecanismo de control de acceso al medio en redes IEEE 802.11. Todo el código generado en el marco del presente trabajo se encuentra publicado en un repositorio de GitHub de acceso libre [13].

Se buscó que la solución propuesta se pudiera adaptar al funcionamiento tradicional de CSMA/CA y que lo complementara, realizando cambios de configuración en tiempo real. En particular, entre los parámetros que se pueden configurar, se seleccionó el intervalo de ventana de contención (CW) que eligen las STAs, debido a que había sido estudiado con éxito en trabajos previos [14–16] y a su fácil manipulación (el valor de dicho parámetro es comunicado por el AP a todas las STAs periódicamente). Si bien existen antecedentes de la aplicación de DRL al control de CW, los mismos presentan deficiencias importantes en su planteo, que lo llevan a un desempeño subóptimo e incluso inestable, como se verá más adelante [17]. Por otro lado, los enfoques tradicionales como el uso de modelos analíticos funcionan bien solo bajo ciertas suposiciones y configuraciones cuasiestáticas o están limitados en su capacidad de generalización.

Para el desarrollo del método E-CCOD, en primer lugar, se puso foco en redes donde todos los dispositivos eran de versiones anteriores del estándar (dispositivos *legacy*). En esta clase de redes no participa el nuevo acceso al medio de 802.11ax, por lo que solo debe controlarse un juego de valores de CW. Se construyó entonces un agente de DRL capaz de seleccionar el valor de CW óptimo frente a distintas configuraciones de la red. Se probaron escenarios donde la cantidad de clientes se mantuvo constante y otros donde la misma fue variando a lo largo de la simulación, se trabajó con los protocolos de transporte UDP y TCP, y con envíos en sentido uplink (UL), downlink (DL) y bidireccional (UL+DL); además, se diseñaron pruebas donde se varió el tipo de tráfico cursado durante la simulación y el agente tuvo que reaccionar y ajustar la CW en consecuencia. Es importante adelantar que el desempeño alcanzado con el método E-CCOD en los escenarios *legacy* evaluados fue muy satisfactorio, logrando seleccionar valores de CW cercanos a los óptimos y obteniendo valores de throughput elevados en todos los casos.

Habiendo aplicado técnicas de DRL a la optimización del control de acceso en redes con dispositivos *legacy* de manera exitosa, se pasó a trabajar con redes "mixtas", es decir, compuestas por dispositivos 802.11ax y *legacy* al mismo tiempo. En la evaluación del estado de situación respecto de la nueva enmienda de IEEE

802.11 se constató que las versiones disponibles al momento de los simuladores de red no representaban fielmente el nuevo acceso al medio. Por este motivo, para poder trabajar con redes mixtas, fue necesario implementar los envíos más eficientes que permite 802.11ax con las funcionalidades disponibles en el simulador (e.g. fragmentación y agregación de paquetes). Luego, se extendió al agente de DRL para que pudiera seleccionar dos juegos de valores de CW (uno para los dispositivos *legacy* y otro para los de la nueva versión) y se lo puso a prueba en un ejemplo de aplicación. A partir de los resultados obtenidos, es posible afirmar la viabilidad de aplicar técnicas de DRL a la optimización del control de acceso en redes donde conviven dispositivos de la nueva versión y de versiones anteriores del estándar. En suma, resulta relativamente sencillo introducir criterios de justicia que permitan garantizar un reparto justo de tiempo de aire entre dispositivos 802.11ax y *legacy*.

1.3. Organización del Documento

El resto del documento se organiza como se indica a continuación. En el Capítulo 2 se realiza un repaso histórico del estándar IEEE 802.11, destacando sus principales características y enmiendas, luego se presenta la última enmienda aprobada y sus nuevas funcionalidades, y por último se analiza el desempeño del estándar. En el Capítulo 3 se introducen los conceptos más importantes de DRL y se presentan algunos de sus algoritmos más populares. En el Capítulo 4 se revisa el trabajo relacionado a la aplicación de técnicas de ML al control de acceso en redes IEEE 802.11, se presenta el trabajo previo tomado como punto de partida para este estudio y se realiza una evaluación de desempeño del método propuesto en él. En dicha evaluación se evidencian serios problemas de diseño e implementación. Estos resultados motivan a que, en el Capítulo 5, se presente un método mejorado, el E-CCOD, para la optimización del control de acceso en redes IEEE 802.11. Luego, se evalúa su desempeño en distintos escenarios de operación realistas y se lo extiende para funcionar en un ejemplo de aplicación de redes con clientes 802.11ax y *legacy* en coexistencia. Finalmente, en el Capítulo 6 se concluye el trabajo.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 2

Evolución del Estándar IEEE 802.11

Este capítulo está dedicado al estándar IEEE 802.11. En primer lugar, se realiza un repaso histórico del mismo, destacando su propósito original (ser la alternativa inalámbrica de Ethernet) y cómo fue su evolución para posicionarse desde hace ya algunos años como una solución considerada en despliegues de alta densidad y para grandes áreas. Se presenta además el funcionamiento básico del mecanismo de control de acceso, el CSMA/CA, el cual se buscará optimizar mediante DRL en capítulos posteriores. Luego, se pone foco en la última enmienda aprobada, la IEEE 802.11ax, sus principales objetivos y las funcionalidades que se introducen o se extienden de enmiendas anteriores. En particular, se describe el nuevo acceso al medio que opera sobre CSMA/CA y se plantean los desafíos que este cambio trae consigo en términos de reparto de recursos. Finalmente, se discute el desempeño del estándar. Por un lado, se presenta el modelo teórico clásico para analizar el desempeño de redes de dispositivos *legacy*, el cual es clave para comprender las bases del problema a modelar en DRL y cuyos resultados son de suma utilidad para evaluar el desempeño del algoritmo construido. Por otro lado, se presentan los resultados de una evaluación de desempeño propia, realizada en base a simulaciones y a pruebas con equipos comerciales, sobre la nueva versión del estándar y la coexistencia con dispositivos de versiones anteriores. De esta forma, se busca diagnosticar la situación actual tecnológica respecto de la nueva enmienda.

2.1. Repaso Histórico

Para la elaboración de estándares, la IEEE crea lo que se conoce como *grupos de trabajo* [18]. En el caso del estándar IEEE 802.11 este grupo se creó en 1990, luego de algunos hechos previos relevantes, tales como la habilitación de bandas de radio para uso industrial, científico y médico (ISM) por la FCC (Federal Communications Commission) en 1985 [19]. El objetivo del grupo IEEE 802.11 era definir un protocolo análogo al IEEE 802.3, más conocido como Ethernet, pero con tecnologías de comunicación inalámbrica. Luego de varios años de trabajo, en 1997 se aprobó la primera versión del estándar, conocido como 802.11-1997 (hoy en día llamado 802.11-legacy, actualmente obsoleto). Esta primera versión soportaba

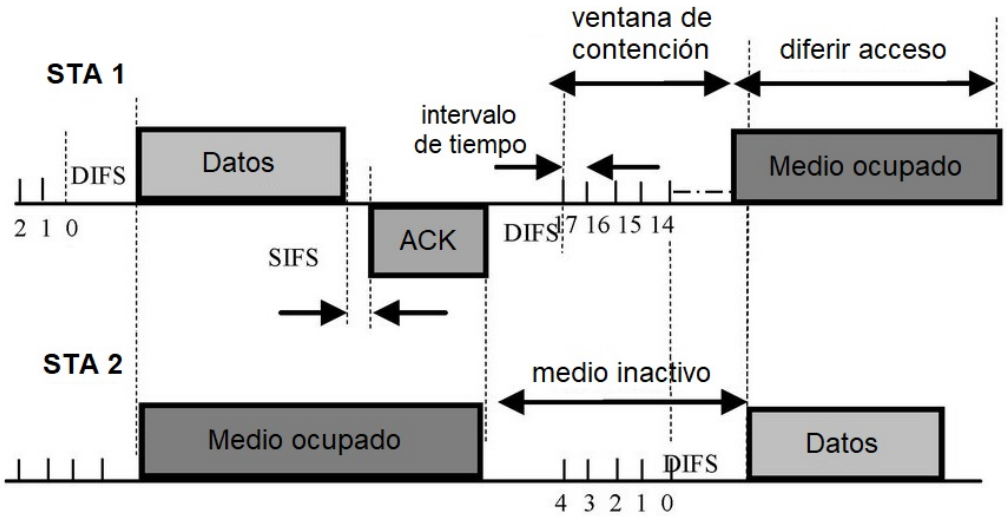

Figura 2.1: Diagrama de tiempos del protocolo CSMA/CA (adaptada de [21]).

velocidades de capa física de 1 y 2 Mbps, en base a tres tecnologías: señales infrarrojas, espectro expandido por secuencia directa (DSSS) o por salto de frecuencia (FHSS). A nivel de control de acceso al medio, ya aparecía desde esa primera versión el mecanismo CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) para evitar colisiones, el cual continúa en las versiones actuales.

Con CSMA/CA, las STAs deben esperar un tiempo aleatorio (*backoff*) antes de transmitir sus tramas, el cual es elegido uniformemente en el rango $[0, CW - 1]$, donde CW refiere a la ventana de contención. Este tiempo se va decrementando a medida que el canal está libre y, al llegar a 0, se efectúa la transmisión. En el primer intento, el valor de CW se establece como el menor tamaño de ventana (CW_{min}); luego, si no se recibe una confirmación para la trama transmitida (debido a una colisión en el receptor o a un canal ruidoso), este valor se duplica hasta alcanzar el tamaño máximo (CW_{max}) [20]. La Figura 2.1 muestra cómo es el proceso de acceso al medio cuando existen dos STAs que deben transmitir. Si bien el mecanismo CSMA/CA es de implementación simple y robusto frente a cambios en la red, no es eficiente, sobre todo en escenarios de alta densidad, que son típicos en la realidad actual. Esto es debido a que en este tipo de escenarios las colisiones son frecuentes, por lo que se desaprovecha mucho tiempo de aire; por el contrario, si se trabaja con valores de CW muy elevados para evitar las colisiones, se está subutilizando el tiempo de aire destinado a transmitir datos [8].

Continuando con la evolución de IEEE 802.11, en 1999 se estandarizaron lo que serían las primeras versiones realmente populares del estándar: 802.11b y 802.11a. La primera, definida para la banda de 2,4 GHz, mantenía la capa física basada en DSSS y alcanzaba un tasa máxima de transmisión de datos de 11 Mbps. La segunda, definida para la banda de 5 GHz, introdujo una capa física basada en OFDM (Orthogonal Frequency-Division Multiplexing), alcanzando tasas de hasta 54 Mbps. Ese mismo año, la Wi-Fi Alliance creó la marca registrada Wi-Fi, basada

Tabla 2.1: Principales enmiendas del estándar IEEE 802.11.

Estándar IEEE	a	b	g	n	ac	ax
Año aprobación	1999	1999	2003	2007	2013	2021
Frecuencia [GHz]	5	2,4	2,4	2,4 & 5	5	2,4 & 5
Tasa bits máx. [bps]	54 M	11 M	54 M	600 M	1,3 G	9,6 G

en el estándar IEEE 802.11. Esta organización sin fines de lucro, es la encargada de la certificación de equipos, lo cual asegura la interoperabilidad entre fabricantes [22]. Fue recién en 2003, cuando se definió la nueva versión del estándar IEEE 802.11g, que se extendió el uso de OFDM también a la banda de 2,4 GHz, siendo esta nueva versión el análogo a 802.11a en dicha banda.

Estos primeros estándares, en particular los de la banda de 2,4 GHz (802.11b y 802.11g), llevaron a la popularización de las redes Wi-Fi, que tuvieron un crecimiento exponencial en la década del 2000. Un protocolo que originalmente estaba pensado solamente como sustituto inalámbrico de Ethernet, pasó a ser una solución considerada en despliegues de alta densidad y para grandes áreas, llegando a casos con alcance metropolitano. Las nuevas demandas de velocidad de transmisión de datos llevaron a la creación de nuevas versiones del estándar. En 2007 se aprobó el estándar 802.11n, definido tanto para 2,4 GHz como para 5 GHz, introduciendo varias mejoras, tanto a nivel de capa física como de capa de control de acceso al medio (MAC). Finalmente en 2013 se aprobó 802.11ac, definido únicamente para la banda de 5 GHz, y con mejoras principalmente a nivel de capa física, por ejemplo estandarizando el *beamforming* [23], que en 802.11n había quedado como opcional. En la Tabla 2.1 se puede ver un resumen de la evolución del estándar IEEE 802.11, incluyendo la última enmienda que será tratada en la siguiente sección; la línea de tiempo detallada del grupo de trabajo de IEEE 802.11 se encuentra en [24].

2.2. El Estándar IEEE 802.11ax

En mayo de 2013, el grupo de trabajo de IEEE HEW (High Efficiency WLAN), también conocido como TGax [25], comenzó el desarrollo de la nueva enmienda IEEE 802.11ax, aprobada finalmente el pasado 9 de febrero de 2021 [6]. El objetivo de este grupo era trabajar en mejoras a las capas física y MAC definidas en el estándar 802.11, tanto en 2,4 GHz como en 5 GHz, con especial énfasis en: (i) mejorar la eficiencia en el uso del espectro y el desempeño por área cubierta y (ii) mejorar el rendimiento en despliegues reales, tanto en interiores como exteriores, con particular foco en escenarios densos y heterogéneos. Estos se caracterizan por la presencia de fuentes de interferencia y niveles de carga en los APs de moderados a muy altos.

Quizás la principal diferencia en el enfoque del diseño de este nuevo estándar fue considerar cierto aprendizaje de los errores cometidos en el pasado. Varias de las mejoras introducidas en 802.11n y 802.11ac, no tuvieron después un impacto

significativo en el desempeño de redes reales desplegadas en ambientes de alta densidad. Esto se debe a que los cambios requerían condiciones muy exigentes (e.g. valores elevados de relación señal a ruido -SNR-, solo alcanzables en ambientes de laboratorio), las cuales no eran acordes a lo que ocurría luego en la práctica. Fue por ello que, desde el comienzo, el TGax se planteó como objetivo realizar mejoras significativas en despliegues reales de alta densidad.

2.2.1. Nuevas Funcionalidades

En IEEE 802.11ax se definen una serie de nuevas funcionalidades que buscan abordar los desafíos descritos anteriormente. Las más importantes se discuten en esta subsección, en base al excelente tutorial de Evgeny Khorov et al. [7].

Modulación y Formato de Trama

La modulación en 802.11ax no ha sufrido grandes cambios. Como se mencionó, desde 802.11a para la banda de 5 GHz y 802.11g para la banda de 2,4 GHz, OFDM ha sido la tecnología predominante a nivel de capa física; esto sigue siendo así. Los anchos de banda soportados son los mismos que los soportados en 802.11ac, 20, 40, 80, 80 + 80[1] y 160 MHz, aunque se debe recordar que 802.11ax opera tanto en la banda de 2,4 GHz (con canales de hasta 40 MHz) como en la de 5 GHz.

Una diferencia importante es el aumento del largo del símbolo OFDM, que se cuadriplica y pasa a ser de 12,8 μs. Esto significa que la cantidad de portadoras también se cuadriplica (pasa a ser 256) y que la separación entre ellas se divide entre cuatro (queda de 78,125 kHz). El objetivo de este cambio es minimizar el impacto de la fluctuación del retardo entre las transmisiones UL multiusuario (MU). Además se puede elegir entre tres valores de intervalo de guarda (GI): 0,8, 1,6 y 3,2 μs, donde se configurará el menor en ambientes con baja densidad de usuarios, permitiendo aprovechar un mayor porcentaje del tiempo de símbolo para carga útil.

Se conservan todas las constelaciones para modular disponibles en versiones anteriores del estándar, agregándose de manera opcional 1024-QAM, que aumenta la tasa de bits pico en condiciones de alta SNR (ver Figura 2.2). No se definen nuevas tasas de codificación, manteniéndose 1/2, 2/3, 3/4 y 5/6. En [26] se muestran las tasas de bits posibles para un flujo espacial (SS) simple, según la modulación y el esquema de codificación (MCS) escogidos. De ella se desprende que la tasa máxima es de 9,6 Gbps, que resulta de utilizar la constelación 1024-QAM con tasa 5/6, canal de 160 MHz, 8 SS y GI de 0,8 μs.

Por otro lado, se definen 4 tipos de trama de capa física: para la transmisión de usuario único (SU), para la transmisión SU de rango extendido (diseñada para una entrega robusta), para la transmisión UL MU y para la transmisión DL MU. Todos estos tipos utilizan la estructura de trama que se muestra en la Figura 2.3 como base, la cual se amplía con campos seleccionados según el tipo específico que se trate. Es importante mencionar que, tanto en las transmisiones UL MU

[1] 80 + 80 corresponde al uso de dos canales de 80 MHz disjuntos.

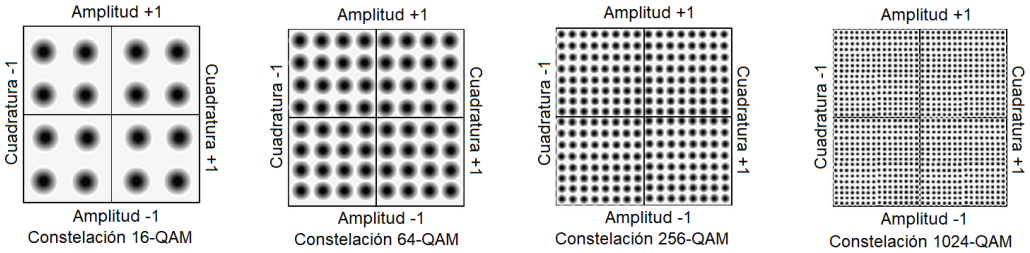

Figura 2.2: Diagramas de constelación para 16-, 64-, 256- y 1024-QAM (adaptada de [27]).

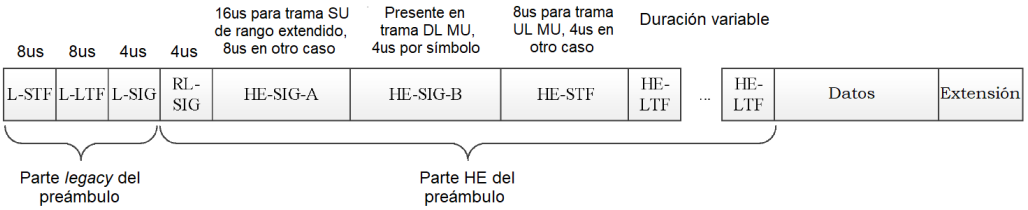

Figura 2.3: Formato de trama de capa física de 802.11ax (adaptada de [7]).

como DL MU, se envía un preámbulo común y, a continuación, la porción de datos dirigida desde o hacia las STAs.

OFDMA sobre CSMA/CA

Seguramente el cambio más importante en la nueva versión del estándar sea el mecanismo de acceso al medio. Ahora pasa a ser el AP quien orquesta el acceso al medio, es decir que la infraestructura de red toma el control de la asignación de recursos, de manera análoga a lo que ocurre en LTE (Long Term Evolution). En concreto, se plantea un mecanismo híbrido entre OFDMA (Orthogonal Frequency-Division Multiple Access) y el tradicional CSMA/CA, que se mantiene por retro-compatibilidad. Como se vio, este último no es eficiente, sobre todo en escenarios de alta densidad; por este motivo se introduce OFDMA, que busca incrementar la proporción de tiempo de aire usado para transmitir datos.

En 802.11ax se agrega la división de múltiples usuarios en el dominio de la frecuencia. Hasta el momento, el canal de Wi-Fi era dividido en varias portadoras OFDM y, en un instante dado, se asignaban todas ellas a una sola STA. Ahora, con OFDMA, grupos individuales de portadoras son asignados individualmente a las STAs (ver Figura 2.4). Estos grupos pueden estar formados por 26, 52, 106, 242, 484, 996 o 2996 portadoras y son denominados Resource Units (RUs). Es importante comentar que la asignación de RUs es válida únicamente por el tiempo que dura una trama.

El empleo de OFDMA está previsto tanto en sentido DL como UL. En el primero, dado que es solo el AP quien transmite, el único mecanismo importante a considerar es que en el preámbulo de la trama se debe indicar la cantidad y tamaño de RUs usados y qué usuarios están asignados a cada RU (junto con sus

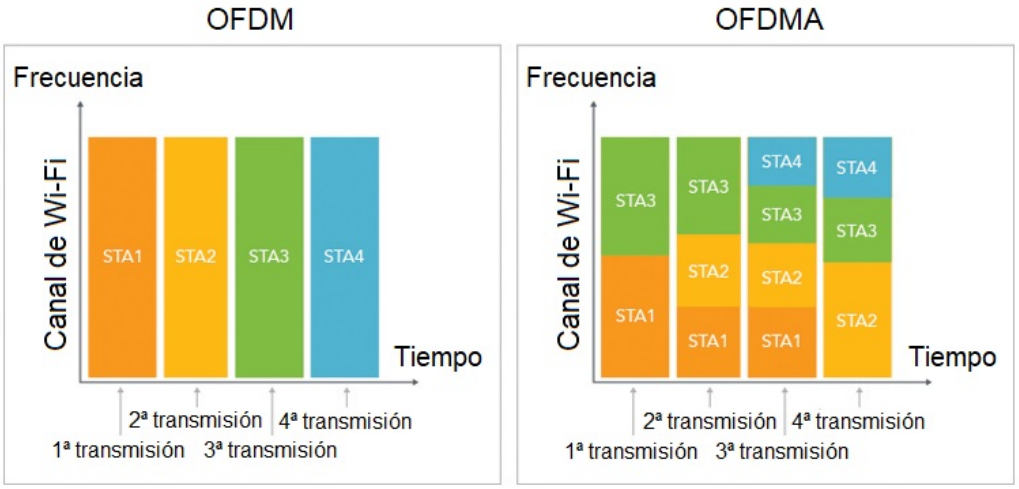

Figura 2.4: Comparación entre transmisiones UL sobre OFDM y OFDMA. Puede verse que con OFDMA la asignación de recursos es mucho más granular (adaptada de [28]).

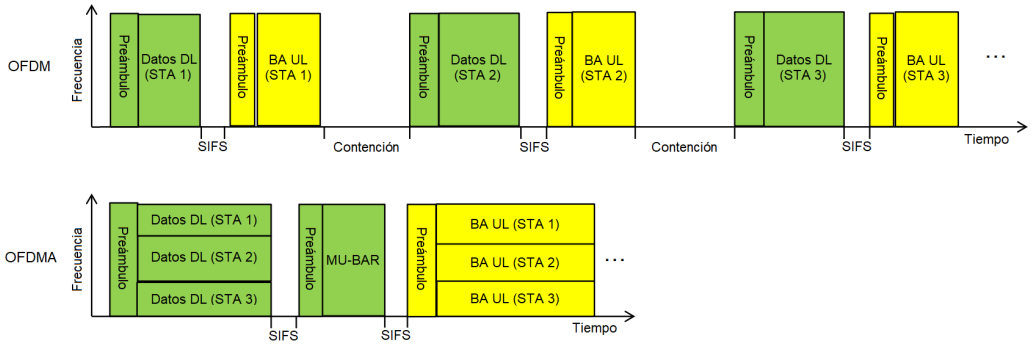

Figura 2.5: Diagrama de tiempos simplificado para transmisiones DL sobre OFDM y OFDMA. Puede verse que con OFDMA el tiempo de aire es utilizado de forma mucho más eficiente (adaptada de [30]).

parámetros para demodular las señales). En la Figura 2.5 se compara el uso del tiempo de aire en varias transmisiones DL, utilizando OFDM junto a CSMA/CA y utilizando OFDMA, de donde queda claro que en este último caso el uso del tiempo es sensiblemente más eficiente[2].

En sentido UL, la implementación es más compleja, puesto que los terminales deben sincronizar sus transmisiones para que puedan ser demoduladas por el AP. Para esto, se utiliza un nuevo tipo de trama denominado Trigger Frame (TF), que es enviado por el AP y define la lista de STAs que van a usar el canal, los parámetros comunes (e.g. GI) y los específicos de cada STA (e.g. MCS). Luego de

[2]BA en la figura significa Block Acknowledgment, una función de capa MAC que aumenta el rendimiento utilizando una sola trama para reconocer múltiples tramas recibidas [29]. BAR refiere a BA Request, una trama enviada por el AP para solicitar el reconocimiento simultáneo de todas las tramas enviadas a las STAs [27].

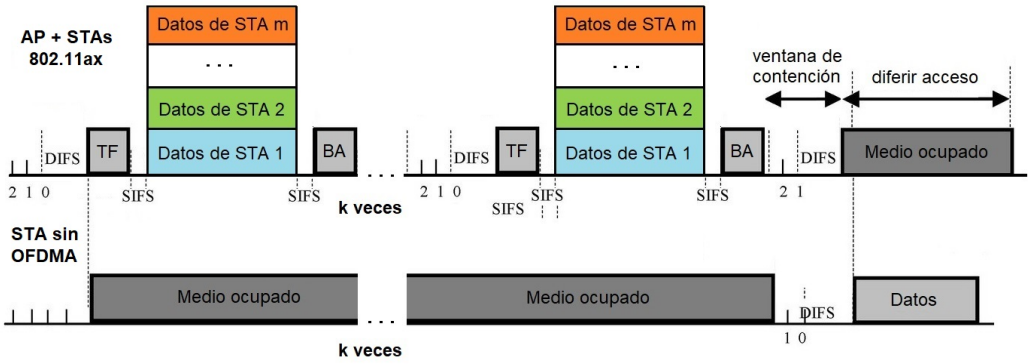

Figura 2.6: Diagrama de tiempos del protocolo CSMA/CA adaptado para su operación con OFDMA. En el primer eje se muestran las tramas DL enviadas por el AP para solicitar a las STAs sus transmisiones (TF) y para reconocerlas (BA), así como las tramas UL MU de datos de las STAs 802.11ax. Por otro lado, en el segundo eje se representa una STA que intenta acceder al medio de forma tradicional, la cual lo hace con mucha menor probabilidad que el AP 802.11ax.

recibir una trama TF, todos los terminales aguardan un tiempo corto (SIFS) y envían sus datos en los RUs que les fueron asignados.

Como se mencionó, el mecanismo descrito anteriormente funciona sobre CSMA/CA, por lo que tanto las STAs como el AP deben competir por el acceso al medio. Sin embargo, para conseguir los beneficios de un uso eficiente del canal, el acceso más usado debe ser OFDMA y no aleatorio y, por tanto, el AP debe tener un acceso prioritario para realizar el envío de tramas TF. Para esto se prevé utilizar 2 juegos de CW_{min} y CW_{max}: uno bajo, utilizado por el AP para ganar el acceso al medio la mayoría del tiempo, y uno elevado utilizado por todas las STAs asociadas para que ganen con mucha menor probabilidad el acceso de forma aleatoria. La Figura 2.6 representa este proceso de acceso al medio donde se prioriza al AP sobre las STAs. En esta selección de valores de los parámetros debe evitarse que los dispositivos *legacy*, que no implementan OFDMA, queden injustamente excluidos del uso del canal o que los APs de determinados fabricantes prioricen las transmisiones de sus propios dispositivos o de fabricantes asociados. Precisamente estos son los desafíos que trae consigo el mecanismo de acceso al medio propuesto en IEEE 802.11ax y que no están resueltos en el estándar, por lo que existe la oportunidad de proponer mecanismos complementarios que los aborden, tal como se plantea en este trabajo.

MU-MIMO

La tecnología MIMO, que consiste en el uso de múltiples antenas transmisoras y receptoras, puede emplearse para mejorar el nivel de señal recibida, pero también para mantener varios flujos de datos en paralelo. En 802.11n estos flujos espaciales se mantenían entre el AP y una única STA, lo que se conoce como SU-MIMO. A partir de 802.11ac se incluyó MU-MIMO, que refiere al envío de varios flujos en

paralelo desde el AP hacia distintas STAs.

En 802.11ax se incorpora MU-MIMO en ambos sentidos: DL y UL. En DL, el AP debe determinar que las condiciones de multicamino le permiten enviar tramas solo a una o a un grupo de STAs dentro de la duración de una trama, sin que las mismas interfieran a otras STAs. Para esto, el AP realiza operaciones de sondeo (*sounding*), que consisten en el envío de tramas nulas por todas sus antenas a todas las STAs asociadas a él. Luego, las STAs responden con medidas de la potencia recibida en cada par de antenas AP-STA. Gracias al uso de OFDMA, los múltiples envíos de medidas pueden hacerse de forma simultánea, reduciendo significativamente el tiempo requerido para este proceso, respecto de la versión anterior del estándar. Otra mejora es que ahora se permiten envíos simultáneos hacia 8 STAs como máximo, duplicando el máximo establecido en 802.11ac.

Por otro lado, MU-MIMO en sentido UL es una funcionalidad que se introduce en esta versión del estándar. Ésta es más simple que la versión para DL, pues no es necesario realizar operaciones de sondeo ni preprocesar las señales para alcanzar a los terminales que, en principio, están a distintas distancias del AP. No obstante, se requiere que las transmisiones de las STAs estén sincronizadas, para lo cual se utiliza el mismo mecanismo de control definido para OFDMA en UL: el envío de tramas TF por parte del AP.

Reducción de Consumo de Potencia

Uno de los objetivos de 802.11ax es aumentar el rendimiento de las redes Wi-Fi manteniendo los requisitos de potencia sin cambios o incluso mejorándolos. Reducir al máximo la potencia requerida es fundamental para los dispositivos móviles, que emplean baterías para su funcionamiento, sobre todo para dispositivos de internet de las cosas (IoT). Para ello, se incorporan mecanismos que permiten a las STAs pactar con el AP intervalos de inactividad (Target Wakeup Time y Opportunistic Power Save) y la posibilidad de que las STAs solo decodifiquen las tramas dirigidas a ellas (Micro Sleep).

Reutilización del Espacio

Como se indicó anteriormente, la nueva versión del estándar busca aumentar el rendimiento de las redes Wi-Fi teniendo en cuenta escenarios de alta densidad. En este tipo de escenarios es común encontrar varios APs, gestionados individualmente o en conjunto, operando en superposición. Para estos casos, se definen mecanismos para reducir la interferencia entre celdas (coloración y umbral de detección y ajuste del nivel de potencia) y hacer un uso eficiente del espectro (*puncturing* de canales y mejoras de AP virtuales).

2.3. Desempeño de IEEE 802.11

2.3.1. Modelo Teórico Legacy

El desempeño de IEEE 802.11 ha sido un tema de estudio muy abordado en la academia desde hace más de dos décadas. Si bien existen varios trabajos relacionados [31–35], uno de los que realizó los aportes más importantes en esta temática fue el de Bianchi [14]. En éste se propone un modelo analítico simple y muy preciso para estimar el throughput saturado de la red, es decir, el throughput total cuando todas las STAs siempre tienen un paquete listo para ser enviado (e.g. cuando las STAs envían flujos UDP a una tasa suficientemente alta). A continuación se discutirá brevemente este modelo clásico.

Se establecen una serie de hipótesis bajo las cuales el modelo funciona de manera razonable. En primer lugar, se considera un escenario ideal: un número fijo de STAs, n, dispuestas alrededor de un AP, de tal manera que todas ellas pueden escuchar a todas las otras STAs de la red. Asimismo, se trabaja con un canal sin pérdidas, se asume que todas las STAs tienen la misma configuración y se incorpora la llamada *hipótesis de desacoplamiento*. Ésta indica que, para cada STA, el evento de que al transmitir ocurra una colisión tiene una distribución Bernoulli de parámetro p, independiente de la historia de las colisiones hasta el momento y del resto de las STAs. Además, p es el mismo para todas las STAs.

A partir de esta última hipótesis, es posible analizar a cada STA por separado. Es así que se define la cadena de Markov de tiempo discreto $\{s(t), b(t)\}$, donde cada STA es un vector de dos dimensiones, formado por su estado de *backoff* y el contador de *backoff*. Los tiempos discretos t y $t+1$ corresponden al inicio de dos intervalos de tiempo consecutivos. Cabe mencionar que, en este modelo, un intervalo de tiempo se define como el período en que el contador de *backoff* puede cambiar su valor. Esto implica que no todos los intervalos de tiempo son de la misma duración, pudiendo incluir, por ejemplo, el tiempo de transmisión de un paquete.

Dadas estas definiciones, se escriben las principales ecuaciones del modelo que permiten estimar el throughput saturado de la red. En primera instancia, se define el parámetro τ como la probabilidad de transmitir en cualquier intervalo de tiempo, o sea, la probabilidad de que la cadena esté en un estado con el contador de *backoff* igual a cero (i.e. $\tau = P(b(t) = 0)$):

$$\tau = \frac{2(1-2p)}{(1-2p)(CW_{min}+1) + pCW_{min}(1-(2p)^m)}, \tag{2.1}$$

con m, tal que $CW_{max} = 2^m CW_{min}$.

Luego, se puede calcular el parámetro p como la probabilidad de que cualquiera de las otras $n-1$ STAs transmita en el mismo intervalo de tiempo (es decir, la probabilidad de colisión) de la siguiente manera:

$$p = 1 - (1-\tau)^{n-1}. \tag{2.2}$$

Resolviendo el sistema de ecuaciones (2.1) y (2.2), se determina el valor de τ, y con éste, se puede escribir P_{tr}, la probabilidad de que haya al menos una

transmisión en el intervalo de tiempo considerado:

$$P_{tr} = 1 - (1 - \tau)^n. \tag{2.3}$$

También se escribe P_s, la probabilidad de que ocurra una transmisión exitosa, que está dada por la probabilidad de que exactamente una STA transmita, condicionada a que al menos una STA transmita:

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{tr}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}. \tag{2.4}$$

Entonces, se puede expresar el throughput S como el cociente entre la carga útil transmitida en un intervalo de tiempo y el largo del intervalo, esto es:

$$S = \frac{P_s P_{tr} E[P]}{(1-P_{tr})\sigma + P_{tr}P_s T_s + P_{tr}(1-P_s)T_c}, \tag{2.5}$$

donde $E[P]$ es el tamaño promedio de la carga útil del paquete, T_s es el tiempo promedio de una transmisión exitosa, T_c es el tiempo promedio de una colisión y σ es la duración de un intervalo de tiempo vacío.

Los tiempos de transmisión exitosa y de colisión se calculan de esta manera:

$$\begin{cases} T_s = H + E[P] + SIFS + \delta + ACK + DIFS + \delta \\ T_c = H + E[P^*] + DIFS + \delta \end{cases}, \tag{2.6}$$

con H la suma de los encabezados de capa física y capa MAC del paquete, δ el retardo de propagación y $E[P^*]$ la longitud promedio de la carga útil del paquete más largo involucrado en una colisión.

Este modelo fue evaluado mediante el uso de simulaciones en el trabajo [5]. En él se realiza una comparación entre los parámetros resultantes de las simulaciones y los estimados por Bianchi, a medida que varía el número total de STAs, para diferentes tasas de bits. En dicho trabajo se concluye que las predicciones obtenidas de probabilidad de colisión y de throughput saturado con el modelo de Bianchi son muy precisas.

Ventana de Contención Óptima

Si se conoce la cantidad de STAs presentes en la red, n, es posible hallar la CW teórica que maximiza el throughput total en un escenario UDP saturado. Puesto que se desea encontrar un valor específico (y no un intervalo), se considera la simplificación $CW_{min} = CW_{max} = CW$. Entonces, se resuelve el sistema de ecuaciones formado por (2.1) y (2.2) para todos los valores posibles de CW (enteros en $[15, 1023]$) y se selecciona aquel valor que consigue el mayor throughput hallado según (2.5). Para el cálculo de S se utilizaron los siguientes valores de los parámetros: estándar 802.11ax, MCS 11, canal de 20 MHz, 1 SS y 1472 bytes de carga útil de los paquetes. Se varió n entre 5 y 50 con paso 5 y se registraron los resultados obtenidos en la Tabla 2.2, también se calculó el caso $n = 1$. De la misma puede verse que la probabilidad de colisión y el throughput se mantienen

Tabla 2.2: Caso UDP saturado. Valores obtenidos mediante modelo de Bianchi. CW óptima y parámetros asociados para distintas cantidades de STAs.

n	CW^*	τ	p	$S[Mbps]$
1	15	0,125	0	42,80
5	34	0,057	0,210	43,75
10	71	0,028	0,224	43,12
15	109	0,018	0,227	42,92
20	146	0,014	0,229	42,82
25	184	0,011	0,230	42,76
30	222	0,009	0,230	42,73
35	259	0,008	0,231	42,70
40	297	0,007	0,231	42,68
45	334	0,006	0,232	42,66
50	372	0,005	0,232	42,65

constantes conforme aumenta n si se configura la CW óptima (CW^*).

Uno de los objetivos del presente trabajo de investigación es el de optimizar el mecanismo de acceso al medio de IEEE 802.11, a partir de seleccionar el valor de CW tal que maximice el throughput de la red (CW óptima). El modelo analítico desarrollado en los párrafos anteriores es de suma relevancia para comprender, al menos para un caso ideal con dispositivos *legacy*, qué parámetros son clave para modelar este problema en términos de DRL, es decir, de qué parámetros del entorno depende el valor de CW a seleccionar por el algoritmo de DRL. De este análisis se puede afirmar que para configurar el valor de CW óptimo, se debe considerar la probabilidad de colisión y la cantidad de STAs de la red. Adicionalmente, los valores de CW óptima y sus parámetros asociados que se presentan en la Tabla 2.2 son de gran utilidad para comparar con los resultados obtenidos por el algoritmo de DRL construido, de forma de evaluar su desempeño para el caso UDP saturado.

Además de funcionar de manera razonable en este caso básico, se requiere que dicho algoritmo lo haga en otros escenarios de operación más realistas (envíos TCP en sentido UL, DL y UL+DL, y variaciones en la cantidad de clientes y en el tráfico cursado), dentro de los cuales se incluye un escenario de coexistencia entre clientes 802.11ax y *legacy*. Por este motivo, es importante comprender el grado de madurez de las implementaciones de 802.11ax disponibles, tanto a nivel de simuladores de red como de equipos comerciales; este tema será tratado en profundidad en la siguiente subsección.

2.3.2. Evaluación de Desempeño de IEEE 802.11ax

Dada la reciente publicación de esta nueva enmienda, es de interés conocer la situación actual tecnológica de las implementaciones en los equipos comerciales y

en las versiones disponibles de los simuladores de red. De esta manera se establecen las bases para trabajar en la optimización del mecanismo de acceso al medio en redes mixtas, o sea, redes donde coexisten dispositivos 802.11ax con dispositivos de versiones anteriores, lo que es uno de los principales objetivos de este estudio. Además, esta información es de suma relevancia tanto para la academia como para la industria, y en particular para Ceibal, ya que una migración de las redes de conectividad desplegadas y de los dispositivos terminales a 802.11ax, tarde o temprano será un hecho, por lo que, es vital conocer cuándo y cómo es la mejor manera de hacerlo.

Definición de los Escenarios de Prueba

El TGax ha definido escenarios típicos para ser utilizados en la evaluación del desempeño de las funcionalidades propuestas en 802.11ax y en la generación de resultados para la calibración de simuladores de red. Estas definiciones están recogidas en el documento [36], en el que se indica que un escenario típico para el entorno educativo (escenario 2) consiste en celdas pequeñas y densas, con una separación entre APs de 10 o 20 m y 100 STAs aproximadamente.

Dado que es de especial interés conocer el comportamiento del nuevo estándar dentro de una celda (i.e. salón de clases), en cada prueba se utiliza un solo AP, siendo siempre un dispositivo que soporta 802.11ax. Con respecto a los clientes, se evalúan distintas cantidades de dispositivos en la celda y diferentes proporciones de dispositivos *legacy* y de aquellos que soportan el nuevo estándar. La Figura 2.7 muestra las distribuciones de las STAs y el AP en el plano XY utilizadas; en todos los casos el AP se instala elevado 2 m sobre el nivel de las STAs.

Actividades como la búsqueda de información, la descarga de archivos, la visualización de contenidos multimedia y la navegación web, que se realizan a diario en las aulas, tienen en común su fuerte componente de tráfico DL. Otras actividades como el uso de las plataformas educativas en línea y las videoconferencias, que han cobrado gran protagonismo en los últimos tiempos a raíz de la emergencia sanitaria provocada por el COVID-19, así como el uso de las redes sociales, tienen un perfil de tráfico UL+DL. Además, todas las actividades mencionadas utilizan TCP como protocolo de transporte; de hecho, los investigadores del grupo de trabajo de ingeniería de internet (IETF) estiman que más del 90 % del tráfico IP se transmite a través de este protocolo [37]. De esto se deduce que, en los escenarios de prueba, se debe considerar el tráfico TCP DL y UL+DL. Además, se utiliza un modelo de tráfico saturado y se elige un tamaño de paquete tal que la unidad máxima de transmisión (MTU) sea de 1500 bytes, el valor establecido por IEEE para interoperar con redes Ethernet [38]. La Tabla 2.3 define los parámetros de configuración utilizados para las simulaciones y las pruebas con equipos comerciales.

Pruebas con Simuladores de Red

Se revisaron los simuladores basados en eventos más utilizados tanto en la academia como en la industria, que implementan al menos de forma parcial 802.11ax. Así, se encontraron los que se mencionan a continuación.

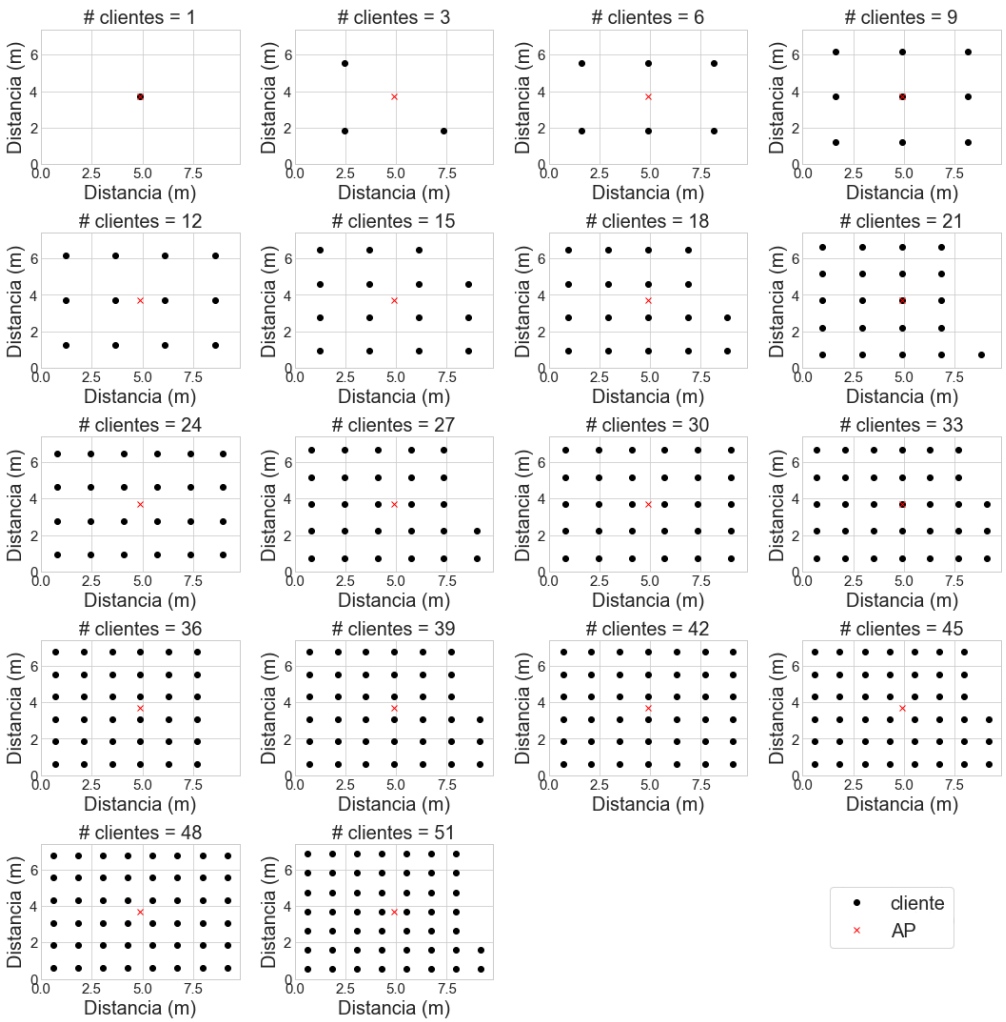

Figura 2.7: Distribuciones de clientes y AP en plano XY utilizadas para la evaluación de desempeño de IEEE 802.11ax en base a simulaciones y a pruebas con equipos comerciales.

ns [39] se refiere a una serie de simuladores de red de código abierto (ns-1, ns-2 y ns-3) ampliamente utilizados en investigación y educación. Gracias al trabajo constante de su comunidad, la versión actual, ns-3, captura muy bien el funcionamiento de las redes IEEE 802.11 y ofrece una gran cantidad de funcionalidades que se pueden incluir en las simulaciones.

Komondor [40] es un simulador de redes inalámbricas de código abierto concebido para contribuir a la investigación de este tema y, por ello, implementa aquellas funcionalidades que no están implementadas en los simuladores tradicionales, como la asignación dinámica de canales o la reutilización del espacio. En particular, uno de los principales objetivos de esta herramienta es simular el comportamiento de las redes 802.11ax, considerando el avance del estándar al 2019.

Tabla 2.3: Parámetros de configuración utilizados para la evaluación de desempeño de IEEE 802.11ax en base a simulaciones y a pruebas con equipos comerciales.

	802.11ax	***legacy***
Potencia tx de AP [dBm]	20	20
Potencia tx de STAs [dBm]	22	22
Frecuencia [GHz]	5	5
Ancho de canal [MHz]	40	40
GI [μs]	3.2	0.8
SS	1	1
MCS	7	7
Sentido	DL, UL+DL	DL, UL+DL
Protocolo de transporte	TCP	TCP
Tamaño de carga útil [bytes]	1460	1460
Modelo de tráfico	saturado	saturado

Resultados Obtenidos. Se realizaron exhaustivas simulaciones con ns-3 para construir la curva de evolución del throughput a medida que aumenta el número de clientes en la celda (de 1 a 51), utilizando las distribuciones presentadas en la Figura 2.7 y los parámetros de configuración definidos en la Tabla 2.3. La Figura 2.8 compara estas curvas para 802.11ax, 802.11ac y 802.11n, utilizando transmisiones DL y UL+DL. Se puede afirmar que 802.11ax no destaca sobre las dos versiones anteriores del estándar, probablemente debido a la no implementación de transmisiones simultáneas con OFDMA en la versión 31 de este simulador (versión actual al momento de realizar las pruebas) [41], lo que permitiría un uso más eficiente del canal.

Además, el gráfico de la Figura 2.9 tiene como objetivo comparar el throughput para 802.11ax en transmisiones UDP DL estimado por ns-3 y por Komondor, ya que las transmisiones TCP y UL no están implementadas en este último. Como se puede observar en el gráfico, con ns-3 se produce una ligera disminución del throughput para los escenarios con mayor número de clientes, algo que no ocurre con Komondor, aunque en este último todos los valores registrados son ligeramente inferiores. La validez de estos resultados fue confirmada por los desarrolladores de Komondor, quienes indicaron además que las diferencias encontradas se deben a diferencias significativas en el modelado de la capa física de los dos simuladores[3].

De las exhaustivas pruebas llevadas a cabo tanto con ns-3 como con Komondor se puede concluir que, al menos en las versiones disponibles en ese momento, existe una inmadurez en la implementación del nuevo estándar, siendo limitados en las funcionalidades ofrecidas. En las versiones de los dos simuladores de red probados no se logró representar fielmente el mecanismo de acceso al medio propuesto en 802.11ax, lo que compromete los resultados obtenidos. Por este motivo, para poder trabajar en la optimización este mecanismo en redes mixtas (como se hará en el

[3]https://github.com/wn-upf/Komondor/issues/155

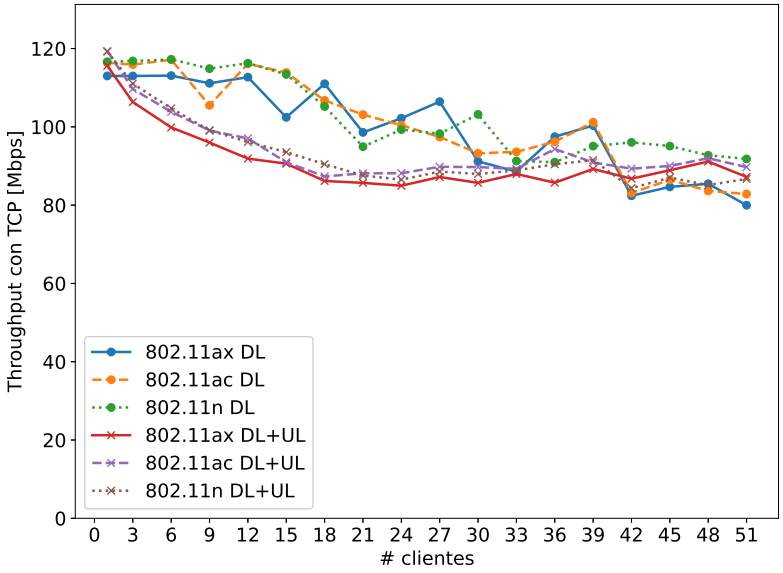

Figura 2.8: Desempeño de IEEE 802.11ax en base a simulaciones. Evolución de throughput para envíos TCP DL y UL+DL, utilizando 802.11n, ac y ax. Gráfico construido con ns-3. Se observa que 802.11ax no destaca sobre las demás versiones probadas.

Capítulo 5), será necesario implementar los envíos MU que permite OFDMA a partir de una abstracción utilizando las funcionalidades que sí están disponibles en los simuladores (e.g. fragmentación y agregación de paquetes).

Pruebas con Equipos Comerciales

En paralelo al trabajo de la IEEE, a medida que el estándar se terminaba de definir con las sucesivas revisiones aprobadas, los fabricantes también iniciaban el proceso de fabricación de nuevos chipsets. Esto llevó a que la Wi-Fi Alliance también comenzara a trabajar en los procesos de certificación, que normalmente corresponden a un subconjunto de las funciones incluidas en el estándar. El proceso de certificación correspondiente, denominado Wi-Fi 6 [42] (análogo a Wi-Fi 5 para 802.11ac, nombrado retrospectivamente), estuvo disponible incluso antes de la aprobación final del estándar; en consecuencia, fue posible adquirir equipos compatibles con Wi-Fi 6, y así se pudo realizar un estudio de desempeño mucho más rico.

Equipamiento Utilizado. Se utilizaron tres modelos de AP para realizar las pruebas, Cisco Catalyst 9115-AXE [43] con antena omnidireccional externa [44], Cisco Meraki MR46 [45] y Aruba 515 [46]. En cuanto a los clientes, se utilizaron laptops Sirio [47] dual-boot (Windows/Ubuntu) entregadas por Ceibal con tarjetas de red 802.11ac Intel 9461NGW [48] y 802.11ax Intel AX200 [49], ambas MIMO 1×1 (o

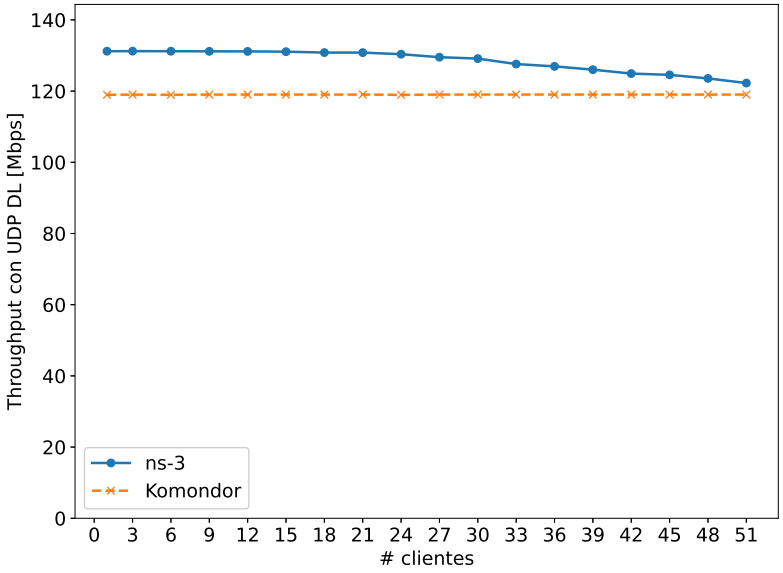

Figura 2.9: Desempeño de IEEE 802.11ax en base a simulaciones. Evolución de throughput para envíos UDP DL, utilizando 802.11ax. Gráfico construido con ns-3 y Komondor. Se observa que con ns-3 se produce una ligera disminución del throughput conforme se incrementan los clientes en la red, algo que no ocurre con Komondor.

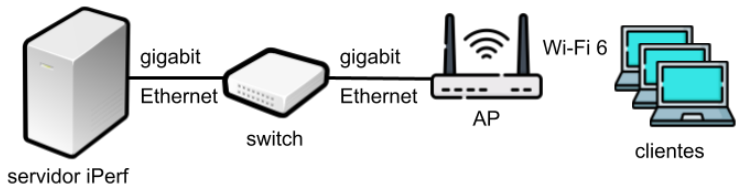

Figura 2.10: Diagrama de maqueta de red montada para la evaluación de desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales.

sea, con una sola antena). También se consideró una laptop Lenovo Thinkpad con Windows y la misma tarjeta de red 802.11ax en modo MIMO 2×2. Además, se utilizó una plataforma de pruebas 802.11n basada en Ubuntu con 32 interfaces de red MIMO 2×2 Intel N-7260 [50], denominada Emulador Wi-Fi [51]. Esta plataforma fue construida en Ceibal para reducir la brecha entre las simulaciones y el mundo real, con el fin de llevar a cabo trabajos de investigación en escenarios típicos de alta densidad.

Para generar tráfico se utilizó la herramienta iPerf [52] y una computadora adicional con una interfaz de red Gigabit Ethernet. El esquema de la Figura 2.10 muestra la maqueta de red configurada para las pruebas.

Resultados Obtenidos. El estudio se dividió en dos grandes etapas. En la primera etapa, denominada *pruebas básicas*, se realizaron pruebas con celdas de 1 AP y hasta 2 clientes y se evaluaron todas las laptops y todos los APs. En la segunda etapa, denominada *pruebas masivas*, se probaron laptops Sirio con tarjeta Wi-Fi 6, el Emulador Wi-Fi y los APs Cisco Catalyst 9115-AX y Aruba 515. En esta etapa, se varió la cantidad de clientes 802.11ax de 1 a 24 para evaluar el desempeño del nuevo estándar por sí solo, comparándolo con los resultados de los clientes 802.11n. Luego, se estudió el desempeño de redes mixtas, variando la relación entre clientes 802.11ax y 802.11n.

Pruebas Básicas. Para las pruebas básicas se tomaron en cuenta los escenarios definidos al comienzo de esta subsección, excepto que en este caso se emplearon canales de 80 MHz, GI de 0,8 μs para 802.11ax (ya que los APs no permitían establecer valores mayores) y de 0,4 μs para 802.11ac, y MCS elegidos dinámicamente. En primera instancia, se probaron individualmente las laptops Sirio con Wi-Fi 5 y Wi-Fi 6 en Windows y Ubuntu y la laptop Lenovo. Posteriormente, se probaron combinaciones de dos laptops formadas por Lenovo y Sirio en sus diferentes sabores (tarjeta Wi-Fi y sistema operativo). Se realizaron transmisiones DL y UL+DL utilizando todos los APs.

De la tabla de MCS en [26] se puede deducir que la tasa de bits más alta a la que se pueden conectar las laptops es de 433,3 Mbps en 802.11ac (canal de 80 MHz, 256-QAM, codificación 5/6 y GI de 0,4 μs) y, utilizando la misma modulación, en 802.11ax es de 480,4 Mbps. Se verificó que las laptops utilizaban un valor de hasta 433,3 Mbps, tanto para Wi-Fi 5 como para Wi-Fi 6. En base a esto, los resultados observados son razonables, con eficiencias (definidas como la relación entre el rendimiento de TCP y la tasa de bits de capa física) siempre por encima de 0,4 y alcanzando valores que superan 0,7 en algunos casos. También vale la pena mencionar que para la laptop Lenovo, que tiene una tarjeta MIMO 2×2, los valores registrados no parecen indicar que esté trabajando con 2 SS.

Con clientes Ubuntu, Wi-Fi 6 tuvo un desempeño promedio de un 15 % mejor que Wi-Fi 5 en todas las pruebas DL y de un 20 % mejor en todas las pruebas UL+DL. Para los clientes Windows, los resultados fueron más divididos, siendo mejores en algunos casos los de Wi-Fi 6 (e.g. Cisco Meraki) y en otros los de Wi-Fi 5 (e.g. Cisco Catalyst). En las pruebas UL+DL se evidenció un fuerte desequilibrio entre UL y DL, siendo en Windows hasta 5 y 10 veces más altas las medidas en UL. También se observó que la laptop Sirio con Ubuntu obtuvo un tráfico significativamente mayor que la Lenovo con Windows.

Pruebas Masivas. Por otro lado, para las pruebas masivas se consideraron los escenarios utilizados para las pruebas básicas, con la excepción de que ahora se trabajó con canales de 40 MHz, como en las simulaciones. En primer lugar, se utilizaron hasta 24 laptops Sirio con tarjeta Wi-Fi 6 y los APs Cisco Catalyst y Aruba para evaluar el rendimiento de 802.11ax con tráfico DL y UL+DL en Ubuntu y Windows. La Figura 2.11 muestra el despliegue realizado para 12 clientes Wi-Fi 6, a modo de ejemplo. Para tener una línea de base para comparar, también se

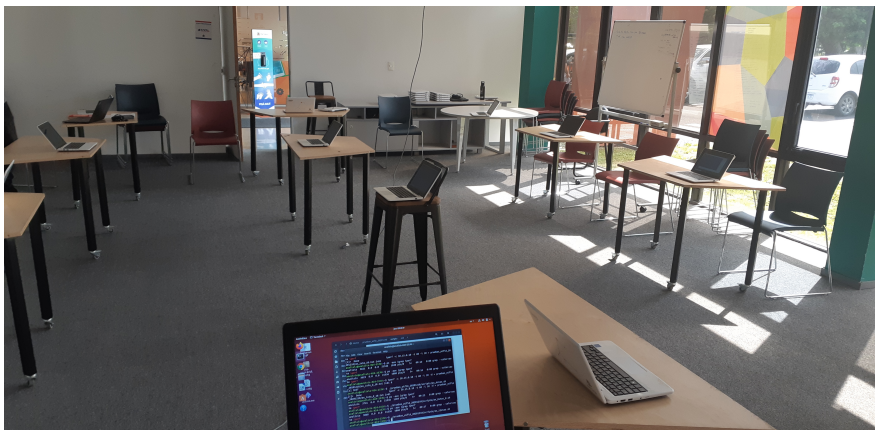

Figura 2.11: Despliegue para 12 clientes Wi-Fi 6 realizado para la evaluación de desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales.

utilizaron hasta 24 interfaces del Emulador Wi-Fi (con Ubuntu) y el AP Cisco Catalyst para evaluar el rendimiento de 802.11n.

Las Figuras 2.12 y 2.13 presentan el desempeño logrado a medida que aumenta el número de clientes para las diferentes configuraciones probadas y envíos DL y UL+DL, respectivamente. Cabe mencionar que con los clientes Ubuntu y el AP Aruba las conexiones eran inestables y los dispositivos después de cierto tiempo se desasociaban, por lo que para este AP solo se presenta el desempeño con clientes Windows.

Con el AP Cisco Catalyst en 802.11ax, se observa una caída muy pronunciada del throughput DL conforme se aumenta el número de clientes, algo que no se vio en las simulaciones (ver Figura 2.8) y que tampoco ocurre con el mismo AP en 802.11n. La caída en la capacidad total de la celda fue significativamente menor en el caso UL+DL, respecto del caso DL. Con el AP Aruba, si bien se observa una caída, ésta es mucho menor, siendo significativa después de que la celda tiene 12 clientes. Con este AP se registraron valores de desempeño superiores a 200 Mbps, aproximadamente 100 Mbps más que con Cisco Catalyst, lo que indica que con Aruba se utilizó una modulación 1024-QAM, según la tabla de MCS de [26]. Tanto este hecho como la mayor escalabilidad a medida que aumenta el número de clientes muestran una mejor implementación del acceso al medio del nuevo estándar en este AP, frente a la del AP Cisco Catalyst; sin embargo, con los clientes Ubuntu, se experimentaron problemas de conexión.

En cuanto a la evaluación de desempeño de las redes mixtas, se utilizó el AP Cisco Catalyst, con diferentes proporciones de laptops Sirio con tarjeta Wi-Fi 6 en Ubuntu y de interfaces 802.11n del Emulador Wi-Fi. Nuevamente, las pruebas se realizaron con tráfico DL y UL+DL. La Figura 2.14 muestra los resultados obtenidos en estas pruebas, mientras que en la Figura 2.15 se vuelven a presentar los resultados registrados en la prueba de referencia con el mismo AP y solo las interfaces del Emulador Wi-Fi como clientes, detallando ahora la proporción de tráfico UL y DL para la prueba UL+DL. Tanto para el caso DL como para el de UL+DL, se puede afirmar que no se obtiene un mayor desempeño agregado

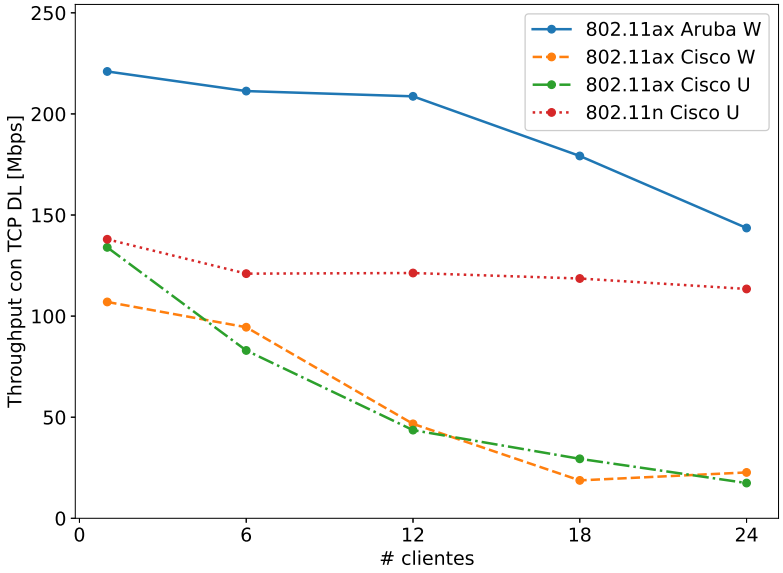

Figura 2.12: Desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. Evolución de throughput para envíos TCP DL, utilizando 802.11n y ax. Se emplearon los APs Aruba [46] y Cisco Catalyst [43], y clientes con Windows (W) y Ubuntu (U). En 802.11ax, se observa una caída del throughput conforme se incrementan los clientes en la red para ambos APs. Por otro lado, los resultados logrados con el AP Aruba son sensiblemente mejores.

cuando se utilizan clientes 802.11ax, en comparación con los clientes 802.11n. En el caso de UL+DL, la capacidad total de la celda se mantiene más estable y se observa un cambio significativo en la relación UL/DL, algo que ya se había notado en las pruebas básicas y que no ocurre con 802.11n (donde predomina el tráfico DL sobre el UL). Cuanto mayor es la proporción de clientes 802.11ax, mayor es la proporción de tráfico UL sobre el DL, lo que parece ser debido a la implementación del proceso encargado de la asignación de recursos (*scheduler* de RUs).

Las pruebas exhaustivas realizadas con equipos comerciales revelan una inmadurez en las implementaciones de 802.11ax por el momento. Por un lado, se detectó el problema con los clientes Ubuntu y el AP Aruba, donde luego de algunos clientes en la celda las conexiones se volvían inestables y los mismos se desasociaban. Por otro lado, los resultados con el AP Cisco Catalyst no fueron satisfactorios, siendo peores que los registrados con 802.11n. Si bien en las pruebas básicas se habían visto mejoras en el rendimiento de Wi-Fi 6 sobre Wi-Fi 5 en algunos casos, a medida que la red comienza a escalar, el rendimiento cae significativamente. No obstante, es de esperar que estos problemas se resuelvan a medida que la tecnología madure. Se estima que este proceso puede tomar algunos años mientras los fabricantes liberan sucesivas actualizaciones de hardware y software de red, y algún tiempo más mientras los usuarios finales recambian sus dispositivos terminales. Más allá de que esto con el paso del tiempo se corrija, los desafíos respecto de un reparto

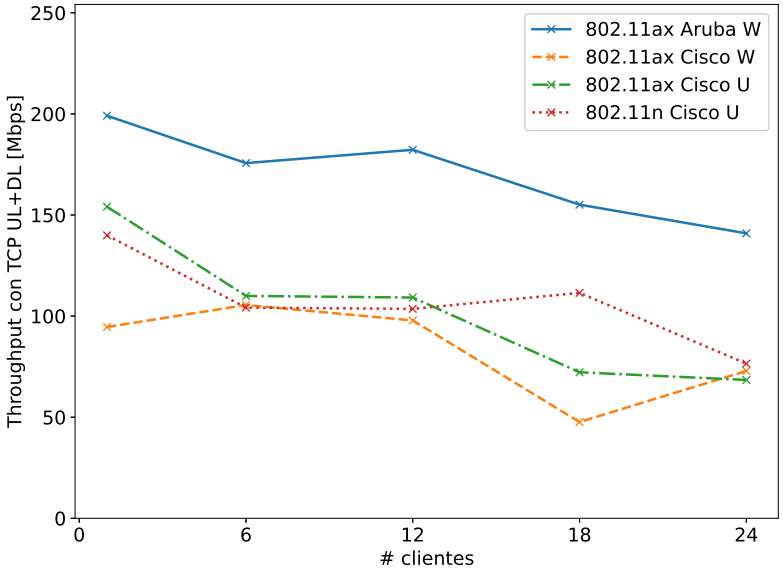

Figura 2.13: Desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. Evolución de throughput para envíos TCP UL+DL, utilizando 802.11n y ax. Se emplearon los APs Aruba [46] y Cisco Catalyst [43], y clientes con Windows (W) y Ubuntu (U). En 802.11ax, se observa una caída del throughput conforme se incrementan los clientes en la red para ambos APs. Por otro lado, los resultados logrados con el AP Aruba son sensiblemente mejores.

de recursos óptimo y justo entre dispositivos de distintas versiones del estándar persisten.

En este capítulo se presentó el estándar IEEE 802.11, su evolución a lo largo del tiempo y, en particular, los objetivos y nuevas funcionalidades de la última enmienda aprobada, la IEEE 802.11ax. Además, se introdujo el mecanismo de acceso al medio tradicional (utilizado en todas las versiones del estándar) y el nuevo mecanismo propuesto en 802.11ax, el cual funciona por encima del primero para asegurar la retrocompatibilidad. Se plantearon los desafíos existentes en torno al mecanismo tradicional (conseguir una operación óptima en escenarios de alta densidad), así como también los desafíos adicionales que surgen a raíz de la aprobación de la nueva enmienda y de la inminente etapa de transición en la que las redes tengan dispositivos 802.11ax y *legacy* en coexistencia (garantizar un reparto justo de recursos entre dispositivos de estos dos tipos). Todos estos desafíos serán abordados en capítulos posteriores, mediante la presentación de un método basado en DRL para la optimización del control de acceso en redes IEEE 802.11. Finalmente, se discutió el desempeño del estándar. Por un lado, se presentó el modelo teórico clásico para analizar el desempeño de redes de dispositivos *legacy*, el cual es clave para comprender las bases del problema a modelar en DRL y cuyos resultados son de suma utilidad para evaluar el desempeño del algoritmo construido. Por otro

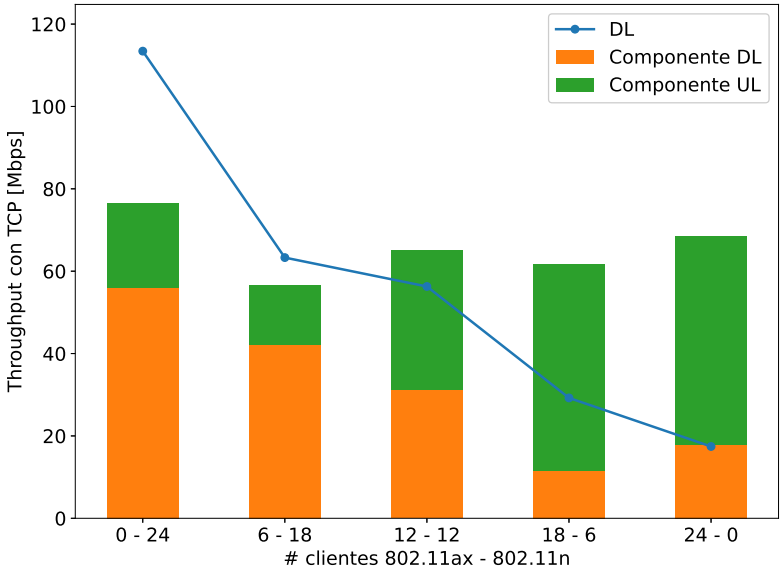

Figura 2.14: Desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. Evolución de throughput y relación UL/DL para envíos TCP DL y UL+DL, en redes mixtas (con clientes 802.11ax y n en coexistencia). Se empleó el AP Cisco Catalyst [43] y clientes con Ubuntu. En DL, se observa una caída pronunciada del throughput conforme se incrementa la proporción de clientes 802.11ax en la red. En UL+DL la capacidad total de la celda se mantiene más estable y se aprecia que a mayor proporción de clientes 802.11ax, mayor es la proporción de tráfico UL sobre el DL.

lado, se exhibieron los resultados de una evaluación de desempeño propia, realizada en base a simulaciones y a pruebas con equipos comerciales, sobre la nueva versión del estándar y la coexistencia con dispositivos de versiones anteriores. Dicha evaluación reveló una inmadurez en las implementaciones de 802.11ax por el momento, tanto a nivel de simuladores de red como de equipos comerciales; sin embargo, es de esperar que esta situación se corrija con las sucesivas actualizaciones de los simuladores y del software y hardware de equipamiento de red y de dispositivos terminales.

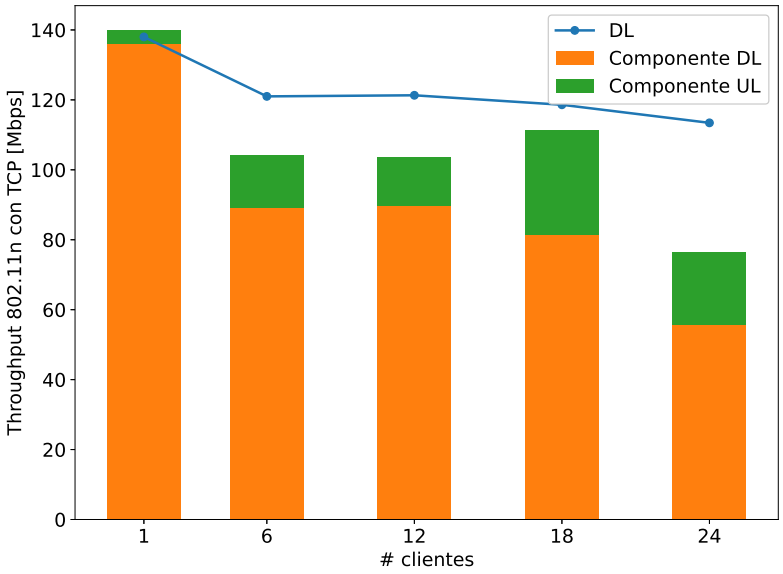

Figura 2.15: Desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. Evolución de throughput y relación UL/DL para envíos TCP DL y UL+DL, utilizando 802.11n. Se empleó el AP Cisco Catalyst [43] y clientes con Ubuntu. Se observa una caída del throughput en el escenario con 24 STAs, la cual es más pronunciada en UL+DL. En todos los casos predomina el tráfico DL sobre el UL.

Capítulo 3

Aprendizaje Profundo por Refuerzo

En este capítulo se presentan los conceptos generales de la inteligencia artificial (AI), el aprendizaje automático (ML), el aprendizaje profundo (DL), el aprendizaje por refuerzo (RL) y el aprendizaje profundo por refuerzo (DRL), todos tópicos muy populares y vinculados. El esquema de la Figura 3.1 podrá ser de ayuda para que el lector comprenda de un vistazo la relación que estos conceptos guardan entre sí. Como se mencionó, existe una tendencia cada vez más creciente en diversas áreas de la industria, y en particular en las telecomunicaciones, respecto de la aplicación de técnicas de ML a la resolución de problemáticas y optimización de procesos y recursos. Dado este contexto, el objetivo principal del presente trabajo es estudiar el potencial de la utilización de DRL para la optimización del mecanismo de control de acceso al medio en redes IEEE 802.11. Para esto, es de suma relevancia profundizar en la teoría detrás de estas técnicas y comprender los fundamentos de sus algoritmos más importantes.

3.1. Inteligencia Artificial y Aprendizaje Automático

La inteligencia artificial (AI) [54, 55] es una disciplina que estudia el análisis y la síntesis de agentes computacionales que actúan de manera inteligente. Un agente computacional es una entidad que realiza acciones en un entorno, cuyas decisiones sobre dichas acciones pueden descomponerse en operaciones primitivas e implementarse en un dispositivo físico. Se dice que un agente actúa inteligentemente cuando toma elecciones apropiadas en base a su experiencia, circunstancias y objetivos, teniendo en cuenta las consecuencias a corto y largo plazo de sus acciones, y adaptándose a entornos y metas cambiantes.

El objetivo central de la AI, desde el punto de vista de la ciencia, es comprender los principios que hacen posible el comportamiento inteligente en sistemas naturales o artificiales. Esto se hace a través del análisis de agentes naturales y artificiales, de la formulación y prueba de hipótesis sobre lo que se necesita para construir agentes inteligentes, y del diseño, construcción y experimentación con sistemas computacionales que realizan tareas comúnmente vistas como que requieren inteligencia. Ejemplos de estas tareas son la traducción de un texto, el reconoci-

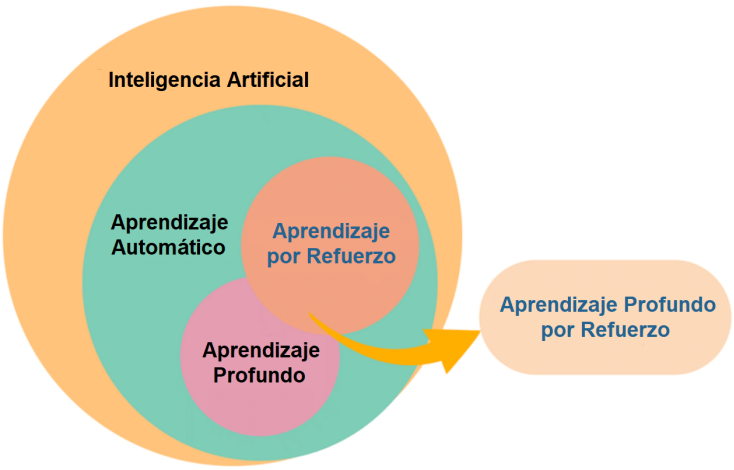

Figura 3.1: Relación entre AI, ML, DL, RL y DRL (adaptada de [53]).

miento de una persona a partir de datos biométricos y el desarrollo de actividades de manera autónoma (vehículos sin conductor y robots o programas que juegan a videojuegos). Por su parte, el objetivo central de la AI, desde el punto de vista de la ingeniería, es el diseño y la síntesis de soluciones de hardware y/o de software, que sean inteligentes y que tengan aplicación en diversas áreas de la industria, como la medicina, las finanzas, las ventas minoristas, el turismo y las telecomunicaciones.

Una de las principales ramas de la AI es el aprendizaje automático (ML) [56,57], la cual se enfoca en el desarrollo y la utilización de sistemas computacionales que pueden aprender y adaptarse sin seguir instrucciones explícitas, mediante el uso de algoritmos y modelos estadísticos para analizar y extraer inferencias de patrones en los datos. En las últimas dos décadas se ha convertido en una herramienta común en casi cualquier tarea que requiera la extracción de información de grandes conjuntos de datos, como ser los motores de búsqueda en línea, los software antispam, las transacciones con tarjetas de crédito protegidas contra fraudes, las cámaras digitales con detección de rostros, las aplicaciones que permiten controlar dispositivos a través de la voz, entre muchas otras. Una característica común de todas estas aplicaciones es que, a diferencia de las aplicaciones informáticas convencionales, debido a la complejidad de los patrones que deben detectarse, no cuentan con una especificación detallada de cómo ejecutar sus tareas. En cambio, emplean algoritmos de ML, los cuales tienen la capacidad de aprender de un conjunto de entradas disponibles (los datos de entrenamiento, que representan la experiencia) y de generar una salida (una o un conjunto de acciones o tareas).

Dichos algoritmos se pueden clasificar en términos generales como no supervisados o supervisados, según el tipo de experiencia que se les permite tener durante el proceso de aprendizaje. Por un lado, los primeros trabajan con un conjunto de datos que contiene, típicamente, muchas características y aprenden propiedades útiles de la estructura de este conjunto de datos. Ejemplos de aplicación de este tipo de algoritmos son la detección de fraude, la detección de software malicioso y la identificación de errores humanos durante la entrada de datos. Por otro lado,

los algoritmos de aprendizaje supervisado emplean un conjunto de datos que contiene características, pero cada ejemplo también está asociado con una etiqueta u objetivo. Luego, la experiencia adquirida durante el entrenamiento se utiliza para predecir esta información faltante en los datos de prueba. Esta clase de algoritmos suele utilizarse en la categorización de textos e imágenes, la detección de rostros, el reconocimiento de firma, la detección de correo no deseado, entre otros. Por último, existen algoritmos de ML que no solo trabajan con un conjunto de datos fijos, si no que interactúan sistemáticamente con su entorno, dando lugar a ciclos de retroalimentación entre el agente y sus experiencias. Estos son los algoritmos de aprendizaje por refuerzo, los cuales se aplican, por lo general, en los juegos, la administración de recursos, los sistemas de recomendaciones personalizadas y la robótica.

3.2. Aprendizaje Profundo

El aprendizaje profundo (DL) [57, 58] es un subconjunto de algoritmos de ML que emplean muchas capas de procesamiento de información no lineal para la extracción y transformación de características, y para el análisis y la clasificación de patrones. La cantidad de capas utilizadas da la *profundidad* del modelo de DL, de allí el nombre de "aprendizaje profundo". Basa su funcionamiento en redes neuronales artificiales, las cuales son modelos simplificados que están inspirados en la manera en que el cerebro humano procesa la información. Están formadas por un número elevado de unidades de procesamiento básicas (*nodos* o *neuronas*) interconectadas, las cuales, por lo general, se organizan en capas. Normalmente, se trata de al menos tres capas: una de entrada, con unidades que representan los parámetros de entrada; una o varias ocultas; y una de salida, con unidades que representan los parámetros de salida. La Figura 3.2 presenta una estructura típica de red neuronal. Es importante aclarar que DL puede aplicarse en aprendizaje supervisado, no supervisado y por refuerzo.

A la salida de cada una de las neuronas existe una función que puede modificar o no el resultado antes de ser entregado a la siguiente. Esta función se conoce como *función de activación* y puede ser lineal (función identidad: $y = x$) o no lineal, aunque suelen utilizarse con mayor frecuencia las de este último tipo, ya que habilitan a resolver problemas más complejos. Las funciones de activación no lineales más populares son las que se muestran en la Figura 3.3. Por su parte, las interconexiones entre los nodos tienen distintos pesos o ponderaciones, los cuales se eligen de forma aleatoria al principio y, con cada predicción realizada en la fase de entrenamiento, se van ajustando hasta alcanzar uno o varios criterios de parada.

Con respecto a su arquitectura, las redes neuronales *prealimentadas* (*feed-forward*) son las más simples. En ellas la información fluye desde la capa de entrada, a través de los cálculos intermedios (las capas ocultas), hacia la capa de salida. No hay conexiones "en el otro sentido", a través de las cuales las salidas se utilicen para retroalimentar el modelo. Según el Teorema de Aproximación Universal [61], una red de este estilo con una sola capa oculta es suficiente para representar cualquier función; sin embargo, su implementación puede no ser viable debido a que

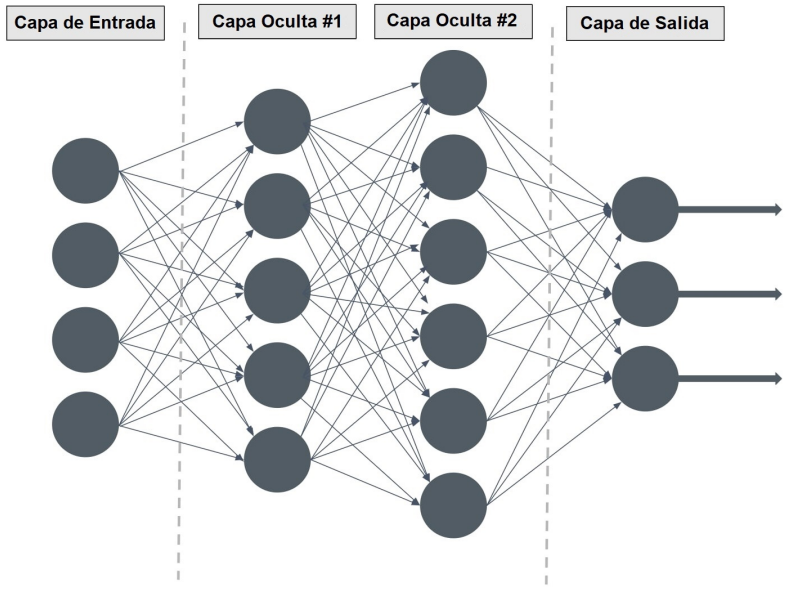

Figura 3.2: Estructura de una red neuronal artificial simple (adaptada de [59]).

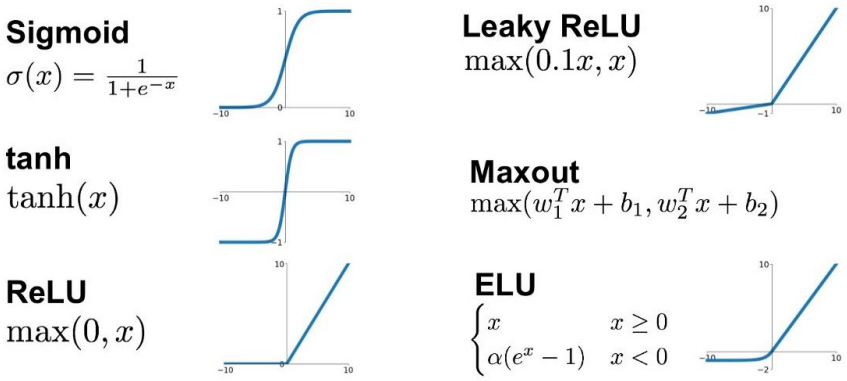

Figura 3.3: Funciones de activación no lineales popularmente utilizadas (extraída de [60]).

la cantidad de neuronas no está acotada (y puede ser inmensamente grande). Es por esto que, en lugar de incrementar el número de neuronas de una sola capa oculta, se suelen incorporar varias capas con un número de neuronas limitado, interconectadas de acuerdo a determinados criterios.

Existen dos tipos de capas utilizados en la mayoría de las redes neuronales, las *totalmente conectadas* (FC) y las *convolucionales* (CNN), las cuales difieren en la forma en que se interconectan las mismas. Por un lado, en las del primer tipo se conecta cada neurona de la capa siguiente con todas las neuronas de la capa anterior, lo que implica tener que calcular una gran cantidad de ponderaciones y puede conducir a problemas de sobreajuste. Sin embargo, son muy buenas para reconocer patrones globales, por lo que, suelen utilizarse como salida de las redes neuronales, para envolver todos los patrones encontrados con las capas anteriores.

Además, el tamaño relativamente pequeño de las capas finales hace que el hecho de tener que calcular los pesos de todas las conexiones no sea un problema. Por otro lado, en las del segundo tipo se conecta cada neurona de la capa siguiente con algunas neuronas de la capa anterior (de acuerdo a la estructura del espacio del problema), así, la cantidad de pesos a aprender será sensiblemente menor que en una capa FC. Las capas CNN son utilizadas para reconocer patrones recurrentes en diferentes capas de entrada, como ser la coincidencia de letras en el procesamiento de texto o de esquinas en el reconocimiento de imágenes.

Como se vio, en las redes prealimentadas la información fluye desde la capa de entrada hacia la capa de salida. Esto implica que la respuesta en un instante dado es independiente de los datos que ha evaluado anteriormente, o sea, son redes sin memoria. En contraposición, las *redes recurrentes* (RNN) hacen uso de información secuencial, por lo que, son ideales para procesar datos en los que existe dependencia con los datos pasados (e.g. análisis de textos, audio o video). De todas maneras, las RNN básicas sufren dos problemas conocidos como *memoria de corto plazo* y *desvanecimiento o explosión de gradientes*. El primero refiere a la limitante de poder almacenar solo un número finito de estados previos; mientras que el segundo ocurre cuando, a medida que se actualizan los pesos de la red, los valores del gradiente se reducen tanto que no llegan a contribuir en los pesos de las capas iniciales, o, por el contrario, se incrementan tanto que provocan que el descenso de gradiente[1] diverja. Estas dificultades son resueltas por las *redes de gran memoria de corto plazo* (LSTM), una extensión de RNN que permite escribir, leer y olvidar información de manera selectiva mediante el uso de compuertas de entrada, de salida y de olvido, respectivamente.

3.3. Aprendizaje por Refuerzo

El aprendizaje por refuerzo (RL) [11, 63] es un enfoque computacional, dentro de ML, que busca comprender y automatizar el aprendizaje basado en objetivos y la toma de decisiones. Consiste en un agente que aprende a partir de la interacción directa con su entorno, sin requerir supervisión o modelos completos del mismo. Dicha interacción se define en términos de estados, acciones y recompensas. Además del agente y el entorno, se identifican tres elementos principales de un sistema de RL: una política, una recompensa y una función de valor. Una política define la forma en que debe comportarse el agente en un momento dado; es decir, es el mapeo entre los estados percibidos del entorno a las acciones que se deben tomar cuando se está en esos estados. Una recompensa establece cuáles son los eventos buenos y malos para el agente, en un sentido inmediato. Por su parte, una función de valor especifica lo que es bueno o malo a largo plazo. En términos generales, el valor de un estado es la cantidad total de recompensa que el agente puede esperar acumular en el futuro, a partir de ese estado; denota el rendimiento esperado.

[1] El descenso de gradiente es un algoritmo que estima numéricamente dónde una función genera sus valores más bajos, es decir que encuentra sus mínimos locales [62].

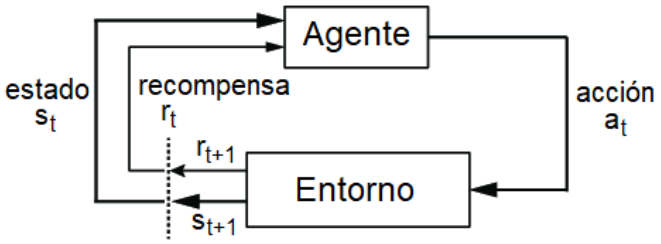

Figura 3.4: Diagrama de interacción entre agente y entorno de RL (adaptada de [64]).

El funcionamiento básico de un sistema de RL es el que se describe a continuación. Un agente observa un estado s_t de su entorno en el instante de tiempo t e interactúa con él tomando una acción a_t. Luego, ambos pasan a un nuevo estado, s_{t+1}, según el estado actual y la acción elegida. El estado es una estadística suficiente del entorno y, por lo tanto, comprende toda la información necesaria para que el agente tome la mejor acción. La mejor secuencia de acciones está determinada por las recompensas proporcionadas por el entorno. Cada vez que el entorno pasa a un nuevo estado, también proporciona una recompensa r_{t+1} al agente como retroalimentación. La Figura 3.4 representa la interacción comentada. El objetivo del agente es aprender una política π que maximice la recompensa acumulada esperada (el rendimiento esperado); cualquier política que cumpla esto será una política óptima (π^*).

Uno de los desafíos que surge en RL es el equilibrio entre exploración y explotación. Para obtener una gran cantidad de recompensas, un agente debe preferir acciones que ha probado en el pasado y que ha encontrado efectivas para producir recompensas. Sin embargo, para descubrir tales acciones, tiene que probar acciones que no ha seleccionado antes. El agente tiene que *explotar* lo que ya sabe para obtener una recompensa, pero también tiene que *explorar* para hacer mejores selecciones de acción en el futuro. El dilema es que ni la exploración ni la explotación pueden perseguirse exclusivamente sin fallar en la tarea, el agente debe probar una variedad de acciones y favorecer progresivamente las que parecen ser las mejores.

3.3.1. Algoritmos

Los algoritmos de RL pueden clasificarse en basados en funciones de valor o en búsqueda de políticas. Los primeros estiman el valor (rendimiento esperado) de estar en un estado dado; mientras que los segundos no necesitan mantener un modelo de función de valor, en cambio, buscan directamente una política óptima. También existe un enfoque híbrido actor-crítico que emplea ambos conceptos. A continuación se presentan algunos de los algoritmos básicos más importantes.

Q-Learning [65]

Este algoritmo pertenece al primer grupo y se basa en valores Q de los pares *(estado, acción)*, los cuales contienen la suma de todas las posibles recompensas.

En principio, esta suma podría ser infinita en caso que no haya un estado final que alcanzar; además, es posible que no se le quiera dar la misma importancia a las recompensas inmediatas que a las recompensas futuras. Para esto se utiliza el factor de descuento γ que toma valores en el intervalo $[0, 1]$ y permite regular el peso de las recompensas futuras.

Si el agente conociera los valores Q de todos los posibles pares *(estado, acción)*, podría utilizarlos para seleccionar la acción adecuada en cada estado. Puesto que el agente no cuenta con esta información, deberá aproximar lo mejor posible esta asignación de valores Q. Como estos valores dependen tanto de recompensas inmediatas como de futuras, se debe establecer un método que sea capaz de calcular el valor final a partir de los valores inmediatos. Para ello se define la siguiente expresión, conocida como ecuación de Bellman o función de acción-valor:

$$Q^*(s_t, a_t) = E[r(s_t, a_t) + \gamma max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})]. \tag{3.1}$$

La misma indica que el valor de Q óptimo para un par *(estado, acción)* es la suma de la recompensa recibida cuando se aplica la acción y del mejor valor Q descontado (multiplicado por γ) que se puede conseguir desde el estado alcanzado al aplicar esa acción.

Para aproximar este cálculo, al principio del aprendizaje los valores Q se establecen en un valor fijo aleatorio, a continuación el agente va tomando pares de *(estado, acción)* y anota cuánta recompensa recibe en ellos, entonces, actualiza el valor almacenado del valor Q de cada par considerando como ciertas las anotaciones tomadas de los otros pares (algunas de las cuales habrán sido aproximadas en pasos anteriores). En resumen, la idea es utilizar la ecuación de Bellman como una actualización iterativa; puede verse que esto converge a la función Q óptima [66].

Vanilla Policy Gradient (VPG) [67]

Este algoritmo es de búsqueda de políticas. Su funcionamiento consiste en aumentar las probabilidades de las acciones que conducen a un mayor rendimiento y, por el contrario, reducir las probabilidades de las acciones que conducen a uno menor, hasta alcanzar una política óptima. Sea π_θ una política con parámetros θ y $J(\pi_\theta)$ el rendimiento esperado de horizonte finito y sin descontar[2] de la política, el gradiente de $J(\pi_\theta)$ se puede escribir como:

$$\nabla_\theta J(\pi_\theta) = E\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t)\right], \tag{3.2}$$

donde A^{π_θ} es la función de ventaja de la política actual. Dicha función se utiliza para comparar distintas acciones y se calcula como la resta entre la función de

[2]El rendimiento esperado de horizonte finito y sin descontar es la suma de la recompensa desde el estado actual hasta el estado objetivo que tiene un intervalo de tiempo fijo o un número finito de intervalos de tiempo [68].

acción-valor y la función de valor. El algoritmo VPG funciona entonces actualizando los parámetros de la política a través de un método de ascenso de gradiente[3] estocástico, tal como se muestra en la siguiente expresión:

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\pi_{\theta_k}). \tag{3.3}$$

Vanilla Actor-Critic (VAC) [70]

El algoritmo Vanilla Actor-Critic es un método Temporal-Difference (TD)[4] que tiene una estructura de memoria separada para representar la política y la función de valor de forma independiente. La estructura de política se conoce como *actor*, porque se utiliza para seleccionar las acciones, y la función de valor estimada se conoce como *crítico*, porque indica qué tan buenas son las acciones realizadas por el actor.

Después de seleccionar una acción a_t en el estado s_t, el crítico evalúa el nuevo estado para determinar si las cosas han ido mejor o peor de lo esperado. A dicha evaluación se la conoce como error de TD y se expresa de esta manera:

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t), \tag{3.4}$$

donde r_{t+1} es la recompensa recibida al cambiar al estado s_{t+1}, V es la función de valor implementada por el crítico y γ el factor de descuento. Este error de TD es la única salida del crítico e impulsa todo el aprendizaje tanto en el actor como en el crítico, como se representa en el esquema de la Figura 3.5. Si el error es positivo, sugiere que la tendencia a seleccionar a_t debería fortalecerse en el futuro, mientras que si es negativo, sugiere que la tendencia debería debilitarse. Esta información es utilizada para actualizar la política implementada por el actor, usando un enfoque tipo Policy Gradient.

Si bien la aplicación de estos algoritmos tuvo algunos casos de éxito en el pasado [72–75], enfoques anteriores carecían de escalabilidad y estaban inherentemente limitados a problemas de dimensiones relativamente bajas. Estas limitaciones existen debido a su complejidad computacional, de memoria y de muestreo, tal como ocurre con otros algoritmos de ML. No obstante, el auge de DL, basado en el uso de redes neuronales artificiales con gran capacidad de representación y aproximación de funciones, ha proporcionado nuevas herramientas para extender la aplicación de RL a problemas más complejos. La combinación de técnicas de DL y RL dan lugar al campo de ML conocido como aprendizaje profundo por refuerzo (DRL), el cual será abordado con mayor detalle en la siguiente sección.

[3] Análogamente al algoritmo de descenso de gradiente, el ascenso de gradiente estima numéricamente dónde una función genera sus valores más altos, es decir que encuentra sus máximos locales [69].

[4] El aprendizaje TD es una técnica no supervisada en la que el agente aprende a predecir el valor esperado de una variable que ocurre al final de una secuencia de estados. Luego, RL amplía esta técnica al permitir que los valores de estado aprendidos guíen las acciones que posteriormente cambian el estado del entorno [71].

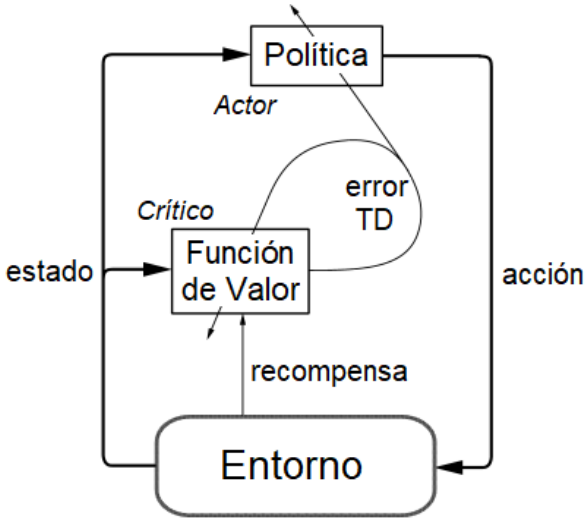

Figura 3.5: Diagrama de interacción entre actor, crítico y entorno en el algoritmo Vanilla Actor-Critic (adaptada de [63]).

3.4. Aprendizaje Profundo por Refuerzo

Como se indicó, el aprendizaje profundo por refuerzo (DRL) [11,76] es un campo de ML que combina técnicas de RL y DL. RL se trata de aprender a partir de la retroalimentación, por ensayo y error; mientras que, DL se trata de aproximar funciones en problemas de alta dimensión. Así, el objetivo de DRL es aprender acciones óptimas que maximicen una recompensa para todos los estados en los que puede estar el entorno, el cual es, por lo general, complejo y de alta dimensión. Para ello, interactúa con este último, probando acciones y aprendiendo de la retroalimentación. Los algoritmos de DRL pueden tomar entradas muy grandes (e.g. los píxeles de sucesivas pantallas de un videojuego) y decidir qué acciones realizar para optimizar un objetivo (e.g. maximizar la puntuación del juego).

3.4.1. Algoritmos

En general, los algoritmos de DRL basan su funcionamiento en algoritmos de RL, los cuales se potencian con la utilización de una o más redes neuronales. A continuación se presentan dos de los algoritmos más populares.

Deep Q-Network (DQN) [66]

Este algoritmo se basa en Q-Learning, introducido en la sección anterior. Q-Learning requiere calcular la función Q de todos los pares *(estado, acción)*. Esto es difícil de escalar, salvo que el sistema tenga espacios discretos muy reducidos de estados y de acciones posibles. En caso que estos espacios sean continuos y de alta dimensión, es más conveniente utilizar DQN, el cual emplea una red neuronal con parámetros θ para estimar los valores Q, esto es, $Q(s,a;\theta) \approx Q^*(s,a)$. Para ello,

se busca minimizar la siguiente función de pérdida en cada paso i:

$$L_i(\theta_i) = E[(y_i - Q(s, a; \theta))^2], \tag{3.5}$$

donde:

$$y_i = r(s_t, a_t) + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_{i-1}). \tag{3.6}$$

Aquí, y_i se denomina objetivo de TD; mientras que $y_i - Q$ corresponde al error de TD, expresado en términos de la función Q. Por su parte, θ_{i-1} representa los parámetros de la red en la iteración anterior.

En la práctica, en lugar de utilizar los parámetros de la última iteración de la red, se suelen emplear los parámetros de una copia de la red original retrasada algunas iteraciones. A esta copia se la denomina *red objetivo*. El uso de este tipo de redes mejora en gran medida la estabilidad en el aprendizaje; en los métodos que no las utilizan, las ecuaciones de actualización de la red son interdependientes de los valores calculados por la propia red, lo que los hace propensos a la divergencia.

Además, se introduce una técnica llamada *repetición de experiencia* para que las actualizaciones de la red sean más estables. En cada paso de la recolección de datos, las transiciones entre estados se agregan a un búfer llamado *búfer de reproducción*. En concreto, la información que se guarda para cada una es la tupla (s_t, a_t, r_t, s_{t+1}). Luego, durante el entrenamiento, en vez de usar la última transición para calcular la pérdida y su gradiente (necesario para minimizar dicha función de pérdida), se utiliza un mini-lote de transiciones muestreado de este búfer. De esta manera, se logra una mayor eficiencia de los datos al reutilizarlos en muchas actualizaciones y una mejor estabilidad al usar transiciones no correlacionadas.

Deep Deterministic Policy Gradient (DDPG) [77]

DDPG es una técnica de DRL que combina Q-Learning con Policy Gradient. Consiste en una evolución al algoritmo VAC y, por tanto, también emplea dos modelos: actor y crítico. El actor decide qué acción debe tomarse y el crítico informa al actor qué tan buena fue la acción y cómo debe ajustarse. Se utilizan cuatro redes neuronales: una red Q con parámetros θ^Q, una red de política determinista con parámetros θ^μ, una red Q objetivo con parámetros $\theta^{Q'}$ y una red de política objetivo con parámetros $\theta^{\mu'}$.

Las redes Q y de política son muy parecidas a un método de actor-crítico simple, donde la primera corresponde al crítico y la segunda al actor. No obstante, en DDPG, el actor asigna directamente los estados a las acciones en lugar de generar una distribución de probabilidad a través de un espacio de acción discreto; de allí el término "deterministic" que compone su nombre. Por su parte, las redes objetivo son copias retrasadas en el tiempo de las redes originales, las cuales se emplean, como se vio en DQN, para mejorar la estabilidad en el aprendizaje.

El funcionamiento básico de DDPG es el que se describe a continuación. Al comienzo se inicializan los parámetros de las redes actor y crítico con valores aleatorios, los de las redes objetivos a partir de los parámetros de las redes originales

y el búfer de reproducción (concepto introducido en DQN). Además, en cada episodio se inicializa un proceso estocástico denominado *ruido de exploración* que se suma a la acción elegida para favorecer el descubrimiento de nuevas acciones. Luego, en cada intervalo de tiempo se selecciona y ejecuta una acción (a partir de la política actual y afectada por el ruido de exploración), y se observa la recompensa y el nuevo estado. Esta transición se almacena en el búfer de reproducción. Por último, se actualizan los parámetros de las redes crítico (minimizando una función de pérdida) y actor (usando Policy Gradient), y los de las redes objetivos (a partir de los parámetros de las redes originales). En el Algoritmo 1 se presenta el pseudocódigo de DDPG.

Algoritmo 1 Pseudocódigo de DDPG (adaptado de [77]).

Inicializar aleatoriamente redes crítico $Q(s, a; \theta^Q)$ y actor $\mu(s; \theta^\mu)$ con pesos θ^Q y θ^μ
Inicializar redes objetivo Q' y μ' con pesos $\theta^{Q'} \leftarrow \theta^Q$ y $\theta^{\mu'} \leftarrow \theta^\mu$
Inicializar búfer de reproducción R
for episodio $= 1, ..., M$ **do**
  Inicializar proceso estocástico η para exploración
  Observar estado inicial s_1
  for t $= 1, ..., T$ **do**
    Seleccionar acción $a_t = \mu(s; \theta^\mu) + \eta$
    Ejecutar acción a_t y observar recompensa r_t y nuevo estado s_{t+1}
    Almacenar transición (s_t, a_t, r_t, s_{t+1}) en R
    Muestrear mini-lote aleatorio de N transiciones (s_i, a_i, r_i, s_{i+1}) de R
    Computar $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}; \theta^{\mu'}); \theta^{Q'})$
    Actualizar red crítico minimizando la función de pérdida:

$$L = \frac{1}{N} \sum_i (y_i - Q(s_i, a_i; \theta^Q))^2$$

    Actualizar red actor usando Policy Gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_i \nabla_a Q(s, a; \theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu} \mu(s; \theta^\mu)|_{s_i}$$

    Actualizar redes objetivo:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1 - \tau)\theta^{Q'}$$
$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1 - \tau)\theta^{\mu'}$$

  end for
end for

En este capítulo se presentaron las bases de DRL y se introdujeron dos de sus principales algoritmos: DQN y DDPG. En particular, existen múltiples trabajos previos que plantean el uso de este último como una herramienta muy útil y eficiente para el manejo de recursos en diferentes áreas de aplicación muy diversas, como ser la comunicación entre vehículos [78, 79], las redes celulares 5G [80, 81] y el control aéreo [82, 83]. En el ámbito de las redes IEEE 802.11 también existen antecedentes de la utilización de DDPG para optimizar la asignación de recursos [17]. No obstante, como se verá en el próximo capítulo, los mismos presentan deficiencias importantes en su planteo, que lo llevan a un desempeño subóptimo e incluso inestable. Estos resultados son los que motivan a proponer, en el presente trabajo, un nuevo método de aplicación de DDPG al control de acceso al medio en redes IEEE 802.11, con el cual se logran obtener resultados satisfactorios en escenarios típicos de la realidad actual.

Capítulo 4

Aplicación de Aprendizaje Profundo por Refuerzo al Control de Acceso en Redes IEEE 802.11

Habiendo realizado un repaso histórico de las principales características y enmiendas de IEEE 802.11 e introducido los principales conceptos y algoritmos de DRL, en este capítulo se explora la aplicación de esta técnica de ML a uno de los mecanismos de 802.11 más importantes, como lo es el control de acceso al medio. El objetivo detrás de esto es su optimización para lograr mejoras significativas a nivel del throughput de la red. En concreto, se presenta, en primera instancia, el trabajo relacionado y se argumenta por qué tiene sentido considerar a DRL como alternativa para abordar este tipo de problemas. Posteriormente, se detalla un estudio previo donde se intentan aplicar dos algoritmos de DRL (DQN y DDPG) a la optimización de la selección de la ventana de contención y se muestra que el mismo tiene graves problemas de diseño e implementación.

4.1. Trabajo Relacionado

Tal como se indicó en la Sección 2.1, el mecanismo de control de acceso al medio utilizado desde las primeras versiones de IEEE 802.11 y hasta la actualidad es el CSMA/CA. Con él, cada STA espera un tiempo aleatorio, elegido en el rango $[0, CW - 1]$, antes de transmitir. Para reducir la probabilidad de que varias STAs seleccionen el mismo número, CW se duplica después de cada colisión, desde CW_{min} hasta CW_{max}. El estándar define valores mínimos y máximos de CW estáticos, lo que resulta en un mecanismo de implementación simple, de baja complejidad computacional y robusto frente a cambios en la red, pero no eficiente y que puede conducir a una operación de la red alejada de la óptima, sobre todo en escenarios de alta densidad [8].

Por esta razón, este tema ha sido objeto de estudio de múltiples trabajos de investigación. Existen enfoques convencionales como el uso de modelos analíticos [14], la aplicación de teoría de control [84] y el monitoreo de usuarios activos

en la red [85], los cuales funcionan bien solo bajo ciertas suposiciones y configuraciones cuasiestáticas o están limitados en su capacidad de generalización. Por otro lado, gracias a las altas capacidades computacionales de los dispositivos de red de la actualidad, es posible considerar métodos de ML para la optimización de este proceso [86–88]; sin embargo, dada la naturaleza del problema, no todos los algoritmos son adecuados. Por ejemplo, los algoritmos de aprendizaje supervisado se basan en minimizar la diferencia entre el resultado inferido y una solución óptima determinada, la cual en este caso no se conoce. Afortunadamente, los algoritmos de RL (y DRL) se adaptan de buena manera al problema de mejorar el desempeño de las redes inalámbricas, ya que se basan en agentes (nodos de la red) que toman acciones (e.g. optimizar parámetros) en un entorno (la red inalámbrica) para maximizar una recompensa (e.g. throughput) [89].

En particular, en un reciente trabajo se propone el método Centralized Contention Window Optimization with Deep Reinforcement Learning (CCOD), para aplicar DRL a la optimización del rendimiento de redes IEEE 802.11 mediante la predicción correcta de los valores de CW [17]. Su implementación está realizada con los algoritmos Deep Q-Network (DQN) y Deep Deterministic Policy Gradient (DDPG), los cuales fueron introducidos en el capítulo anterior, y se exhibe su funcionamiento con redes IEEE 802.11ax (en realidad, con los aspectos de la nueva versión del estándar que están disponibles en la versión 29 de ns-3, que no incluyen funcionalidades importantes como OFDMA [90]). Se consideran dos escenarios de operación, uno **estático**, donde todas las STAs se encienden a la vez y se mantienen activas hasta el final de la simulación, y uno **dinámico**, donde se encienden solo 5 STAs al principio y luego las restantes se van activando de a una hasta el final de la simulación. En el presente trabajo, el análisis de este método se enfoca en la versión con DDPG, entendiendo que éste es un método más avanzado que DQN, con el cual los autores muestran mejores resultados.

El sistema montado se compone de los siguientes elementos:

- El *agente* está ubicado en el AP, ya que este equipo tiene una visión global de la red, puede controlar de manera centralizada las STAs asociadas y cumple con los requisitos computacionales.

- El *estado* del entorno se describe en términos de la probabilidad de colisión de la red, es decir, la cantidad de tramas no recibidas, sobre el total de tramas transmitidas:
$$p = \frac{N_{tx} - N_{rx}}{N_{tx}}. \tag{4.1}$$
Más específicamente, esta probabilidad se expresa como una matriz de dimensión 4×2, donde la primera columna corresponde a la media (μ) y la segunda a la varianza (σ^2) de múltiples muestras. Se parte de un total de 300 muestras, las cuales son divididas en 4 tramos contiguos en el tiempo y para cada tramo se calcula su media y varianza. Con esta información se completa dicha matriz.

- La *acción* consiste en seleccionar un nuevo valor de CW. En realidad, lo que el agente elige es el parámetro α, que toma valores continuos en $[0, 6]$ y

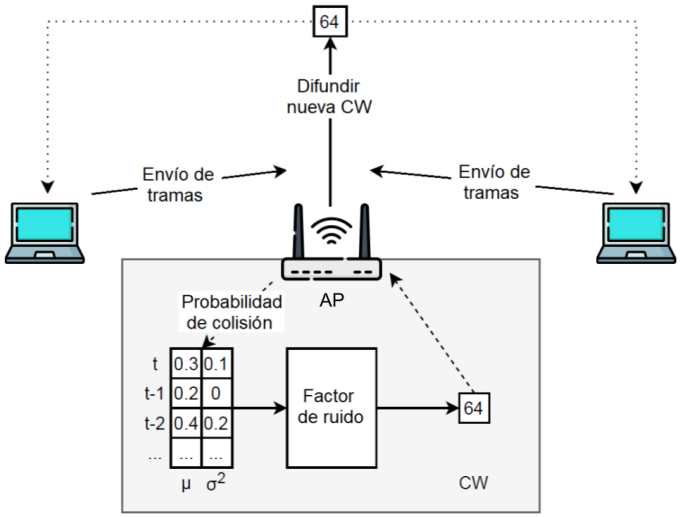

Figura 4.1: Topología utilizada y proceso de actualización de CW en el método CCOD (adaptada de [17]).

luego se configura la CW según la siguiente expresión:

$$CW = \left\lfloor 2^{\alpha+4} - 1 \right\rfloor. \tag{4.2}$$

Para favorecer la exploración, cada acción es modificada por un factor de ruido, que decae conforme se avanza en la etapa de entrenamiento.

- La *recompensa* usada es el throughput de la red, normalizado adecuadamente para que tome valores en $[0, 1]$.

Se considera una topología de red compuesta por un AP y múltiples clientes, en línea con los escenarios educativos trabajados en la Subsección 2.3.2. En esta topología, el agente (ubicado en el AP) calcula la probabilidad de colisión de la red en cada instante de tiempo. Luego, con el procesamiento descrito más arriba, genera la matriz de estado 4×2, la cual es utilizada para seleccionar el nuevo valor de CW (modificado por un factor de ruido). Este nuevo valor de CW es difundido a todos los clientes asociados a la red mediante las tramas Beacon enviadas por el AP. Por su parte, el envío de tramas de datos solo está previsto en sentido UL, con protocolo de transporte UDP y modelo de tráfico saturado. Como se vio en la subsección mencionada, este tipo de envíos (UDP saturado y en sentido UL) no es común en la realidad actual, por esta razón, en el presente trabajo se buscará construir un agente que opere en escenarios más realistas. En la Figura 4.1 se presenta la topología de red utilizada, así como también el proceso de actualización de CW y el envío de tramas de datos.

Por otro lado, se utilizan dos redes neuronales, *actor* y *crítico* (ver Subsección 3.4.1), cuyas arquitecturas son similares entre sí y corresponden a una red LSTM seguida de dos capas FC, resultando en una configuración de $8 \times 128 \times 64$. El diagrama de las redes puede verse en la Figura 4.2.

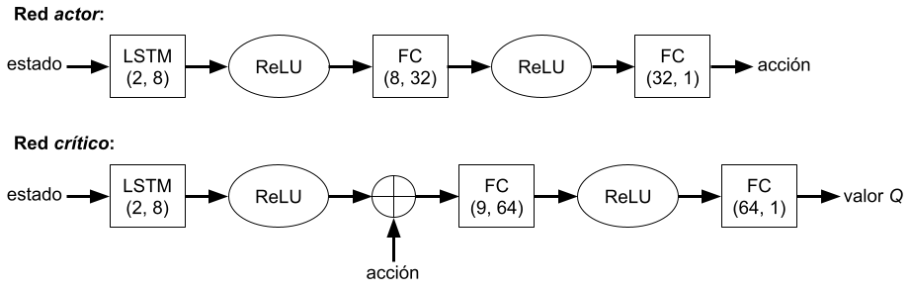

Figura 4.2: Diagrama de redes *actor* y *crítico* involucradas en el método CCOD.

4.2. Desempeño del Agente para el Caso UDP

Como primera evaluación del sistema construido en el trabajo previo [17] se realizaron ejecuciones sin intervención del agente de DRL, considerando un escenario estático donde la cantidad de STAs activas se mantuvo constante durante toda la simulación. Se utilizaron distintas cantidades de STAs (de 5 a 50, con paso 5, también $n = 1$) y la CW fijada en el valor óptimo (ver Tabla 2.2). En el cálculo de probabilidad de colisión, se pasaron a considerar solo los envíos y recepciones de tramas de datos (se excluyeron tramas de control), corrigiendo así valores elevados de esta probabilidad que se registraban en un principio. Además, se modificó el código para identificar los envíos y recepciones a nivel de STA; esta modificación, así como todo el nuevo código generado en el marco del presente trabajo, se encuentra publicado en un repositorio de GitHub de acceso libre [13].

En la Tabla 4.1 se presenta la probabilidad de colisión y el throughput total registrados en las pruebas; por otro lado, en la Figura 4.3 se muestra el reparto de la probabilidad de colisión por STA, su valor promedio y su valor a nivel de red. Analizando estos resultados, puede afirmarse que, en general, los valores de probabilidad de colisión y throughput obtenidos son razonables y están cercanos a los hallados teóricamente. Además, puede verse un reparto parejo de las transmisiones (y de p) entre todas las STAs de la red, registrándose únicamente 3 STAs que no lograron transmitir significativamente para el caso $n = 50$.

Luego de verificar que el funcionamiento de la simulación está cercano a lo esperado en la teoría, es de interés ir un paso más adelante y evaluar la capacidad de aprendizaje del algoritmo construido. Para ello, se ejecutaron pruebas donde se entrenó con una cantidad de STAs en la red y luego se evaluó con otras cantidades, utilizando escenarios estáticos y dinámicos en ambas etapas. En la Tabla 4.2 se resumen los resultados registrados; mientras que, en las Figuras 4.4 y 4.5 se presenta la evolución de la CW, el throughput total, la probabilidad de colisión y la recompensa acumulada de los entrenamientos en ambos escenarios de operación. Adicionalmente, la Figura 4.6 muestra los resultados obtenidos en las evaluaciones en escenarios dinámicos (se omiten los gráficos de evaluación en escenarios estáticos debido a la poca variación de sus parámetros).

A partir de los resultados de las evaluaciones realizadas en escenarios estáticos (ver Tabla 4.2), se puede afirmar que el algoritmo funciona bien solo si se lo

Tabla 4.1: Caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Probabilidad de colisión y throughput registrados cuando se configura CW óptima para distintas cantidades de STAs. Se observa que estos valores son razonables y están cercanos a los hallados teóricamente.

n	CW^*	p	$S[Mbps]$
1	15	0	37,56
5	34	0,206	39,14
10	71	0,236	38,43
15	109	0,343	36,70
20	146	0,251	38,09
25	184	0,312	37,08
30	222	0,362	36,08
35	259	0,232	38,05
40	297	0,262	37,77
45	334	0,289	37,52
50	372	0,299	37,37

Tabla 4.2: Desempeño del método CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. CW aprendida y probabilidad de colisión y throughput registrados en fase de evaluación, utilizando escenarios estáticos (Est.) y dinámicos (Din.). Se observa que el algoritmo selecciona una CW cercana a la óptima solo cuando se lo entrena con una cantidad fija de STAs y luego se lo evalúa con la misma cantidad (fila 3).

Esc.	n_{entr}	n_{eval}	CW^*	CW	p	$S[Mbps]$
Est.	25	10	71	(194; 207)	(0,166; 0,186)	30,84
Est.	25	25	184	(192; 205)	(0,174; 0,211)	38,07
Est.	25	50	372	(160; 187)	(0,317; 0,482)	35,54
Din.	(5; 25)	(5; 10)	(34; 71)	(37; 198)	(0,114; 0,222)	38,35
Din.	(5; 25)	(5; 25)	(34; 184)	(35; 298)	(0,126; 0,263)	37,89
Din.	(5; 25)	(5; 50)	(34; 372)	(35; 273)	(0,131; 0,356)	37,52

evalúa con la misma cantidad de STAs con la que fue entrenado ($n = 25$); en otro caso ($n = 10, 50$), se selecciona una CW alejada de la óptima. Otro hallazgo interesante que se desprende de estas evaluaciones es la relación que existe entre la CW y la probabilidad de colisión: a mayor CW, menor probabilidad de colisión y viceversa (e.g. el pico de CW en el entorno de 20000 s de tiempo de simulación que se corresponde con el valle de probabilidad de colisión en el mismo instante aproximadamente o el valle de CW y pico de probabilidad de colisión cerca de los 60000 s en la Figura 4.4).

Con respecto a las evaluaciones en escenarios dinámicos, es importante recordar que tanto en el entrenamiento como en la evaluación, la cantidad de STAs se va incrementando desde $n_{min} = 5$ hasta el máximo n establecido en cada caso

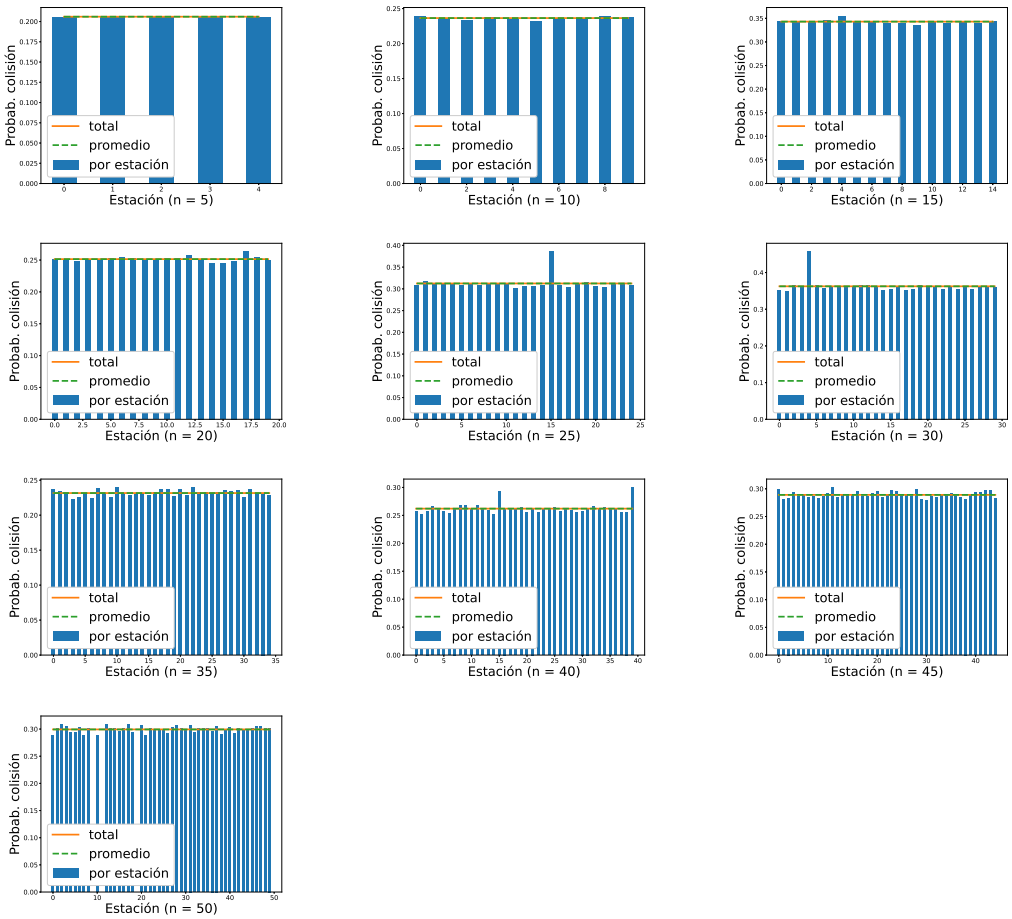

Figura 4.3: Caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Probabilidad de colisión por STA, en promedio y de la red cuando se configura CW óptima para distintas cantidades de STAs. En general, se aprecia un reparto parejo de los valores de esta probabilidad entre todas las STAs de la red.

($n_{max} = 10, 25, 50$ según corresponda). Además, la CW óptima también se incrementa conforme varía n, como se vio en la Tabla 2.2. De dicha tabla se tiene que, sin importar la cantidad de STAs, la probabilidad de colisión se mantiene constante si se configura la CW óptima. Esto no ocurre en estas evaluaciones; puede verse en la Tabla 4.2 y la Figura 4.6 que, conforme se avanza en la ejecución, esta probabilidad tiende a aumentar, pese a algunos comportamientos oscilatorios. En relación a lo anterior, se aprecia que la CW elegida oscila (esto es observado sobre todo para $n = (5; 10)$ y $n = (5; 25)$), lo que es indicio de que el agente no tiene certeza sobre el valor a seleccionar. Finalmente, para el caso $n = (5; 50)$ los valores de CW aprendidos en la segunda mitad de la simulación están muy alejados de los óptimos hallados teóricamente.

A partir del análisis teórico realizado en la Subsección 2.3.1 se había concluido que el valor de CW que maximiza el throughput depende de la probabilidad de

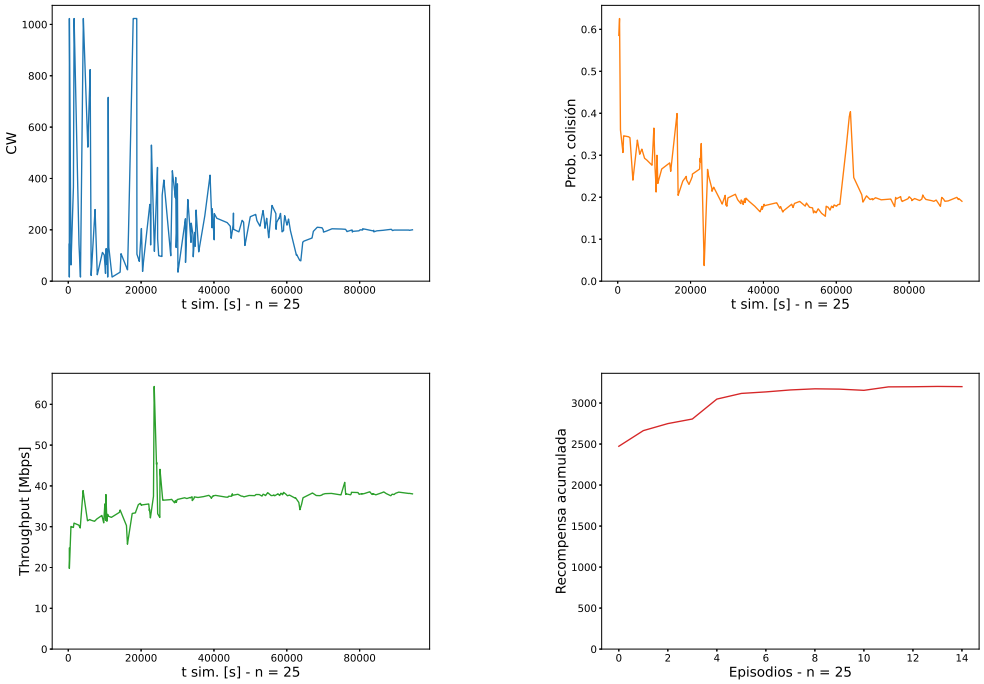

Figura 4.4: Desempeño del método CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario estático con $n = 25$ STAs.

colisión y de la cantidad de STAs de la red. En el modelo propuesto, el entorno se observa únicamente a partir de la probabilidad de colisión, la cual es constante para cualquier cantidad de STAs si se configura la CW óptima, como se mostró en la Tabla 2.2. Por este motivo, el sistema a priori no podría inferir los cambios en esta cantidad y ajustar la CW en consecuencia. Los resultados obtenidos en las pruebas conducidas confirman esta conclusión, ya que el agente solo funcionó bien cuando se lo entrenó con una cantidad fija de STAs y luego se lo evaluó con esa misma cantidad (escenario estático). En otro caso, cuando en la red ocurría un cambio en la cantidad de STAs, el agente no tenía certeza sobre el valor de CW a seleccionar y, por tanto, se observaban valores por encima y por debajo del que se venía utilizando (es decir, el valor de CW oscilaba). El hecho de seleccionar un valor de CW adecuado solo cuando la cantidad de STAs se mantiene constante no agrega valor a la hora de llevarlo a la práctica, ya que, si se conoce previamente esta cantidad, basta con configurar a mano la CW óptima para la misma. Además, que la red mantenga una cantidad fija de STAs durante su operación no es algo que ocurra en la realidad.

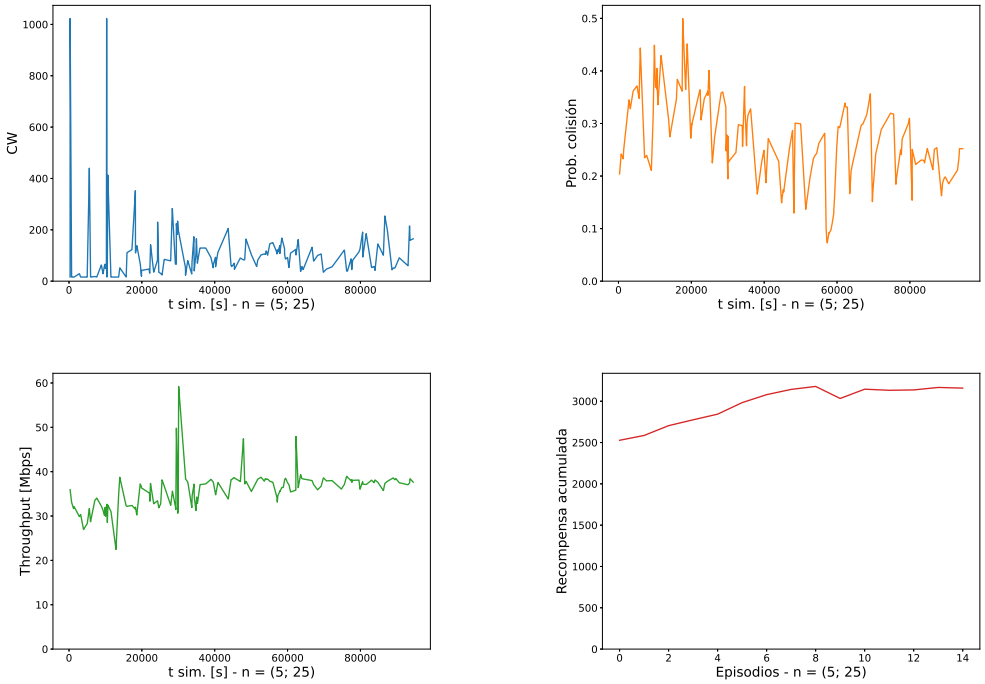

Figura 4.5: Desempeño del método CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario dinámico con $n = (5; 25)$ STAs.

Dados los hallazgos mencionados, en el siguiente capítulo se propone un nuevo método E-CCOD (Enhanced-CCOD), basado en el método CCOD presentado y evaluado en este capítulo, para la optimización del rendimiento de redes IEEE 802.11 mediante la predicción correcta de los valores de CW. Este método mejorado incorpora, entre otras cosas, un estimador de la cantidad de STAs activas en la red como parte del estado del entorno. Como se podrá ver, el agregado de esta información es clave para mejorar sensiblemente el desempeño del agente y poder adaptarse de buena manera a cambios en la cantidad de STAs y en el tipo de tráfico cursado en la red.

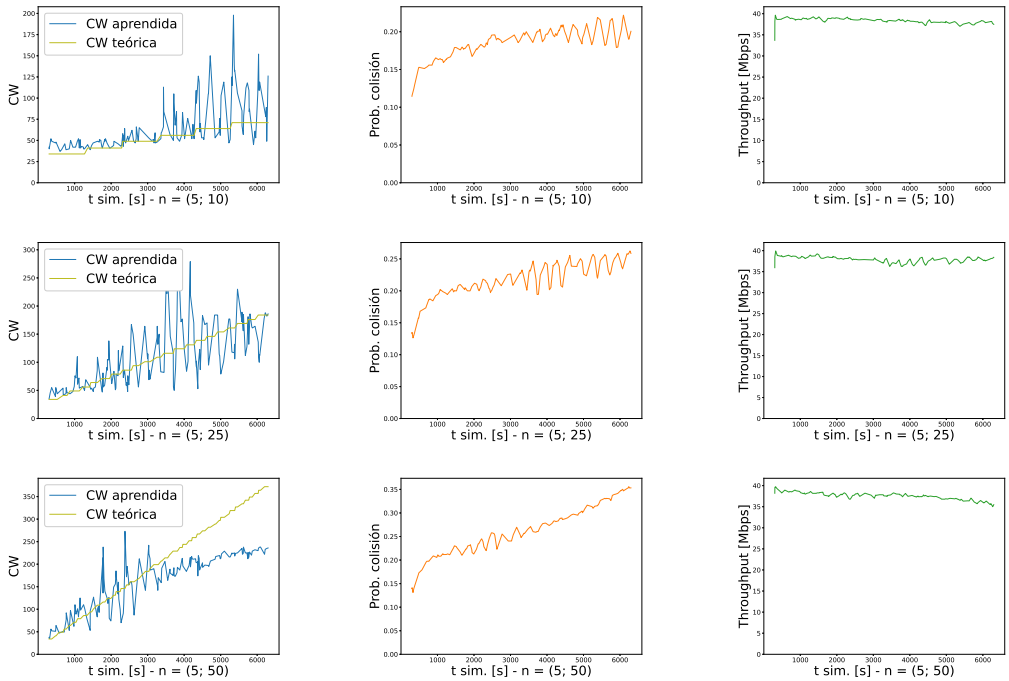

Figura 4.6: Desempeño del método CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión y throughput en fase de evaluación, utilizando escenarios dinámicos. Se emplearon distintas cantidades de STAs: $n = (5; 10)$ - fila 1, $n = (5; 25)$ - fila 2 y $n = (5; 50)$ - fila 3. Se aprecian comportamientos oscilatorios en los parámetros y el algoritmo no logra seleccionar una CW cercana a la óptima en ningún caso.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Presentación y Evaluación del Método E-CCOD para la Optimización del Control de Acceso en IEEE 802.11

En el capítulo anterior se exploraron las bondades de la aplicación de DRL a la optimización de la elección de la CW en redes IEEE 802.11. En particular, se analizó un estudio previo donde se ofrecía un modelado del problema y una implementación del sistema en ns-3. A partir de la evaluación realizada, se concluyó que el algoritmo construido no tiene el desempeño deseado. No obstante, el trabajo fue de gran utilidad, no solo como un primer acercamiento a la temática, sino que también por dejar disponible una implementación del sistema completo: el agente (algoritmo de DDPG aplicado a optimizar la CW), su entorno (simulación de una red IEEE 802.11) y la interacción entre ellos. Cabe señalar además que el agente es capaz de aprender la CW óptima si se mantiene constante la cantidad de STAs, lo cual no es un resultado trivial. Dicha implementación fue tomada como base para proponer un método para la optimización del control de acceso en redes IEEE 802.11 más robusto, el cual consigue mejoras significativas de desempeño.

Así, en este capítulo se presenta el método E-CCOD (Enhanced-CCOD) que permite optimizar el rendimiento de redes IEEE 802.11 mediante la predicción correcta de los valores de CW. Se busca que pueda ser aplicado de manera satisfactoria en escenarios cercanos a la realidad, por lo que, se consideran UDP y TCP como protocolos de transporte, todos los sentidos de envío de tramas (UL, DL y UL+DL) y variaciones en el tráfico cursado. Además, se presenta un ejemplo de funcionamiento del agente en una red donde coexisten clientes 802.11ax con otros de versiones anteriores del estándar, con el objetivo de visualizar los desafíos que surgen en términos de un reparto justo de recursos entre estos tipos de clientes.

5.1. Presentación del Método

Como se vio, el método Centralized Contention Window Optimization with Deep Reinforcement Learning (CCOD) propuesto en [17] solo logra seleccionar la

CW óptima en redes donde se mantiene constante la cantidad de clientes, lo cual no es suficiente para llevarlo a la práctica y está alejado de lo que ocurre en la realidad. En base a estos resultados y a las conclusiones extraídas en el capítulo anterior respecto de la información necesaria para observar el entorno en este tipo de problemas, se pretende analizar el sistema de DRL montado y proponer modificaciones, las cuales serán recogidas en un nuevo método más robusto, denominado Enhanced-CCOD (E-CCOD).

Para ello, en primer lugar, se desacopló el sistema formado por el agente y el entorno basado en simulaciones, y se pasó a utilizar un entorno analítico, basado en las ecuaciones de Bianchi introducidas en la Subsección 2.3.1. De esta forma, se quitó complejidad al sistema a analizar, se ganó agilidad en la ejecución de las pruebas y se hizo foco en el diseño del modelo propuesto.

Luego, se realizaron pruebas análogas a las aplicadas con el sistema basado en simulaciones: se entrenó el sistema en escenarios estáticos y dinámicos, con $n = 25$ y $n = (5; 25)$, respectivamente, y se lo evaluó en ambos tipos de escenarios con $n = 10, 25, 50$ y $n = (5; 10), (5; 25), (5; 50)$. De esta forma, se pudo constatar que el agente solo logra configurar la CW óptima si se evalúa con la misma cantidad de STAs (fija) con la que fue entrenado, tal como ocurría con el sistema original. Además, se concluyó que el tipo de escenario más adecuado para entrenar el agente es el dinámico, ya que, al variar la cantidad de STAs en la red (de $n_{min} = 5$ hasta $n_{max} = 10, 25, 50$ según corresponda), se lo deja preparado para actuar en las múltiples realidades distintas que se pueda enfrentar en la práctica. Habiendo verificado que ambos sistemas se comportan de la misma manera, se siguió trabajando con el que emplea un entorno analítico (cuya principal ventaja es la reducción significativa del tiempo de ejecución de las pruebas), con foco en escenarios dinámicos.

En el modelo actual, el estado del entorno se observa a partir de la probabilidad de colisión de la red. Únicamente con esta información, el agente no puede detectar los cambios en la cantidad de STAs de la red y ajustar la CW en consecuencia, ya que dicha probabilidad es constante para cualquier cantidad de STAs si se configura la CW óptima (ver Tabla 2.2). Entonces, se propone incorporar la cantidad de STAs como otra variable del estado. Para el agente es relevante llevar la cuenta de las STAs activas, n', es decir, las que transmiten una cantidad significativa de tramas; este número puede ser diferente a la cantidad total de STAs, n, e incluso puede variar a lo largo de la operación de la red, por lo que se debe definir un mecanismo para estimarlo.

El mecanismo considerado a nivel de simulaciones es el siguiente. Se define un vector de largo n, inicializado en 0. Luego, en cada transmisión de capa MAC, se incrementa en 1 el valor de la posición correspondiente a la ID de la STA que envió dicha trama. Finalmente, para inferir la cantidad de STAs activas en la red, n', en un instante dado (al momento de medir el estado del entorno), se utiliza un umbral (fijado en 5 tramas), el cual deben superar para que sean consideradas activas. Una vez reportado este valor, el vector se resetea y se comienzan a contar transmisiones para el siguiente instante de tiempo. En la práctica, dado que no se tiene de antemano el valor n, se puede definir un vector de tamaño fijo suficientemente

grande o uno de tamaño variable que se incremente a medida que se detecta una nueva STA. Luego, las IDs de las STAs pueden ser sus direcciones MAC y generarse un mapeo entre estas direcciones y las posiciones del vector.

A partir del funcionamiento del método CCOD e incorporando estas modificaciones comentadas, se puede definir el funcionamiento del método Enhanced-CCOD (E-CCOD), cuyo pseudocódigo se presenta en el Algoritmo 2.

Algoritmo 2 Pseudocódigo del método E-CCOD.

Obtener los pesos del agente θ
Obtener la función de acción del agente A_θ
Definir N_{tx} y N_{rx} como la cantidad de tramas transmitidas y recibidas
Definir n' como la cantidad de STAs activas
Definir *train* como bandera para indicar la fase de operación
Definir Δt como período de interacción
Inicializar búfer de observaciones B
Inicializar búfer de reproducción R
for episodio $= 1, ..., M$ **do**
  Inicializar proceso estocástico η para exploración
  $CW \leftarrow 15$
  $s \leftarrow$ vector de ceros
  for t $= 1, ..., T$ con paso Δt **do**
    $N_{tx} \leftarrow$ cantidad de tramas transmitidas
    $N_{rx} \leftarrow$ cantidad de tramas recibidas
    $n' \leftarrow$ cantidad de STAs activas
    B.push($\frac{N_{tx}-N_{rx}}{N_{tx}}$, n')
    Calcular estado: $obs \leftarrow$ preprocess(B)
    if *train* **then**
      Seleccionar acción: $\alpha_t \leftarrow A_\theta(obs) + \eta$
    else
      Seleccionar acción: $\alpha_t \leftarrow A_\theta(obs)$
    end if
    Ejecutar acción: $CW \leftarrow \lfloor 2^{\alpha_t+4} - 1 \rfloor$
    if *train* **then**
      $Thr \leftarrow \frac{N_{rx}}{\Delta t}$
      Calcular recompensa: $r \leftarrow$ normalize(Thr)
      Almacenar la transición: R.push((obs, a, r, s))
      $s \leftarrow obs$
      $b \leftarrow$ mini-lote de R muestreado
      Realizar optimización de θ basado en b
    end if
  end for
end for

La implementación de dicho método fue realizada utilizando el lenguaje de programación Python [91] y las bibliotecas Pytorch [92] (para modelado en ML) y NumPy [93] (para operaciones algebraicas), entre otras. El entorno se montó haciendo uso del simulador ns-3.29 [90], mientras que para la integración de todo el sistema (agente y entorno) se empleó la herramienta ns3-gym [94]. A grandes rasgos, el agente (ubicado en el AP) observa la cantidad de tramas enviadas (N_{tx}) y recibidas (N_{rx}) para calcular la probabilidad de colisión. Según el sentido de los envíos, una cantidad la puede monitorear directamente y la otra deberá ser enviada por las STAs "a cuestas" (*piggybacking*) con los datos. Además calcula la cantidad de STAs activas (n') con el procedimiento explicado en párrafos anteriores. Con esta información genera una acción, que consiste en seleccionar un nuevo valor de CW, el cual es difundido a todas las STAs asociadas a la red. Consta de dos fases: entrenamiento y evaluación. En la primera fase, las acciones son modificadas por un factor de ruido gaussiano (para favorecer la exploración), el cual decae conforme se avanza en el aprendizaje (para favorecer la explotación). Por su parte, en la segunda fase este ruido no participa y se selecciona en todo momento las acciones que el agente entiende más adecuadas.

Por último, resta verificar si el agente mejorado, que observa el estado de su entorno a partir de la probabilidad de colisión y la cantidad de STAs activas, logra un desempeño satisfactorio. Esto se hizo, en primera instancia, empleando el sistema basado en un entorno analítico, donde se asume que todas las STAs de la red están activas. Para ello, se condujeron pruebas análogas a las realizadas anteriormente, poniendo foco en escenarios dinámicos. La Figura 5.1 presenta los resultados del entrenamiento con $n = (5; 25)$; mientras que la Figura 5.2 los de las evaluaciones con $n = (5; 10), (5; 25), (5; 50)$. En particular, en los gráficos correspondientes a las dos primeras evaluaciones puede observarse que, con los cambios introducidos, el agente acompaña la evolución de la red, seleccionando como CW la óptima (o muy cercana) en cada caso. Además, cabe destacar que ahora los valores de probabilidad de colisión y de throughput alcanzados están dentro de los valores esperados, según la Tabla 2.2, y no presentan los comportamientos oscilatorios que se habían registrado antes (ver Figura 4.6).

Por otro lado, cuando se evaluó con $n = (5; 50)$, el agente no funcionó correctamente, sobre todo en la segunda mitad de la prueba, lo que indica que el modelo no es inductivo, es decir, que no puede generalizar a casos que nunca vio en el entrenamiento. Un resultado positivo, sin embargo, es que a cantidades de STAs mayores a las usadas en el entrenamiento, la CW seleccionada no disminuye. Por lo tanto, si se realiza un entrenamiento hasta la cantidad de STAs máxima que se espera en la red (este número se conoce a priori ya que uno dimensiona su infraestructura teniendo presente la cantidad de clientes potenciales que la utilizarán), o bien, hasta una cantidad de STAs, tal que, se alcance la CW máxima, se estarían cubriendo todos los casos con los que el agente trabajaría en la práctica.

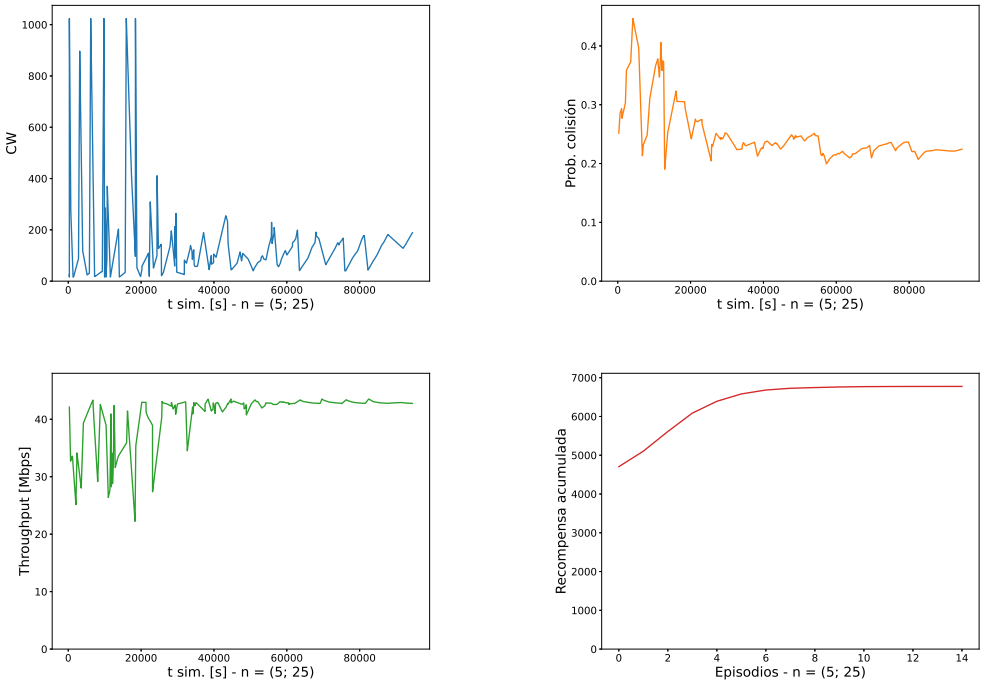

Figura 5.1: Desempeño del método E-CCOD para el caso UDP saturado. Valores obtenidos mediante modelo de Bianchi. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario dinámico con $n = (5; 25)$ STAs.

5.2. Evaluación de Desempeño del Método

En la sección anterior se presentó un método mejorado para la optimización del control de acceso en IEEE 802.11 y se lo puso a prueba en un entorno analítico, basado en las ecuaciones de Bianchi introducidas en la Subsección 2.3.1. Habiendo constatado que, con los cambios incorporados, el desempeño del agente mejora sensiblemente, es de interés ponerlo a prueba en distintos escenarios de operación: caso UDP en sentido UL, caso TCP en sentido UL, DL y UL+DL y caso con variaciones en el tráfico cursado. El primero consiste en el mismo escenario planteado en el trabajo tomado de base y es interesante considerarlo para poder comparar el desempeño de ambos agentes (el del método CCOD y el del método E-CCOD); mientras que, los dos últimos consisten en escenarios cercanos a la realidad. Cabe destacar que para todas las pruebas realizadas en esta sección se emplearon entornos basados en simulaciones.

5.2.1. Caso UDP

Para la evaluación de este primer caso de funcionamiento, se realizaron las mismas pruebas que las aplicadas en la sección anterior: entrenamiento en un escenario dinámico con $n = (5; 25)$ y evaluación en el mismo tipo de escenario con

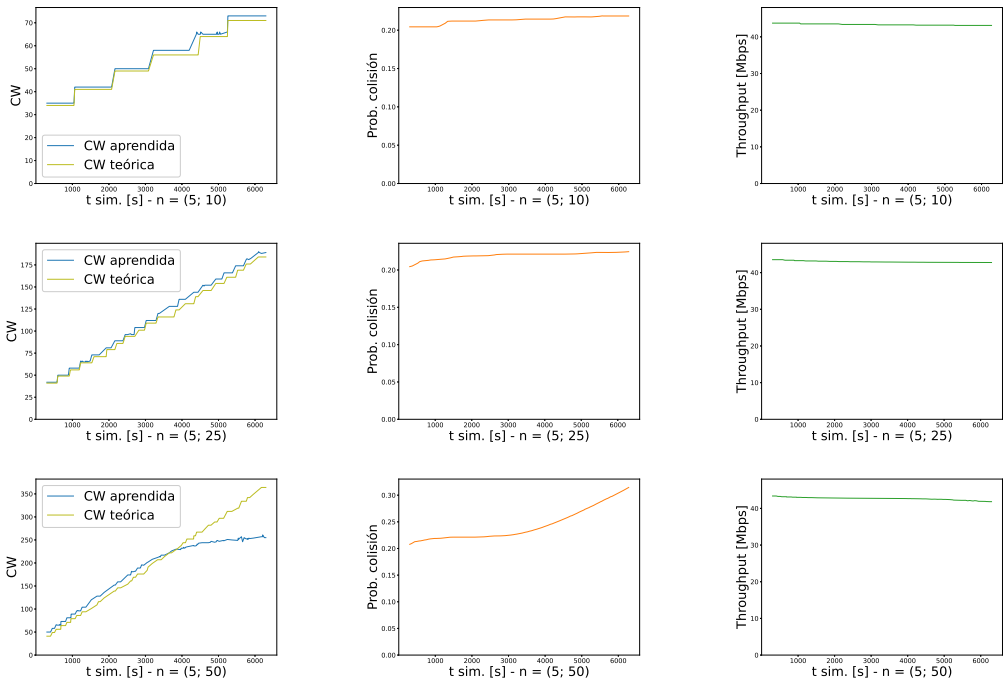

Figura 5.2: Desempeño del método E-CCOD para el caso UDP saturado. Valores obtenidos mediante modelo de Bianchi. Evolución de CW, probabilidad de colisión y throughput en fase de evaluación, utilizando escenarios dinámicos. Se emplearon distintas cantidades de STAs: $n = (5; 10)$ - fila 1, $n = (5; 25)$ - fila 2 y $n = (5; 50)$ - fila 3. Se aprecia que el algoritmo acompaña la evolución de la red y logra seleccionar una CW cercana a la óptima, siempre que se utilicen cantidades de STAs usadas en la fase de entrenamiento.

$n = (5; 10), (5; 25); (5; 50)$, de forma de verificar que el buen desempeño también se obtiene si se utilizan un entorno basado en simulaciones y el estimador de STAs activas. La evolución de los principales parámetros para las fases de entrenamiento y evaluación se muestran en las Figuras 5.3 y 5.4, respectivamente.

A partir del análisis de estas figuras, es posible afirmar que se logran resultados análogos a los obtenidos con el sistema basado en un entorno analítico para $n = (5; 10), (5; 25)$. Se aprecia una leve sobreestimación en la CW aprendida, algo que ya había ocurrido en la prueba con el agente original (sin el agregado de la cantidad de STAs activas como parte del estado) en un escenario estático con $n = 25$ (ver Tabla 4.2); no obstante, ahora el agente logra captar de muy buena manera la dinámica del sistema aumentando el valor de CW al detectar una nueva STA en la red. En suma, vale la pena señalar que el agente no solo selecciona una CW cercana a la óptima para los casos vistos en la etapa de entrenamiento (como ocurría con las pruebas con entorno analítico), sino que también logra extrapolar lo aprendido y responder bien para casos nunca vistos ($n > 25$), aunque lo hace de manera más ruidosa.

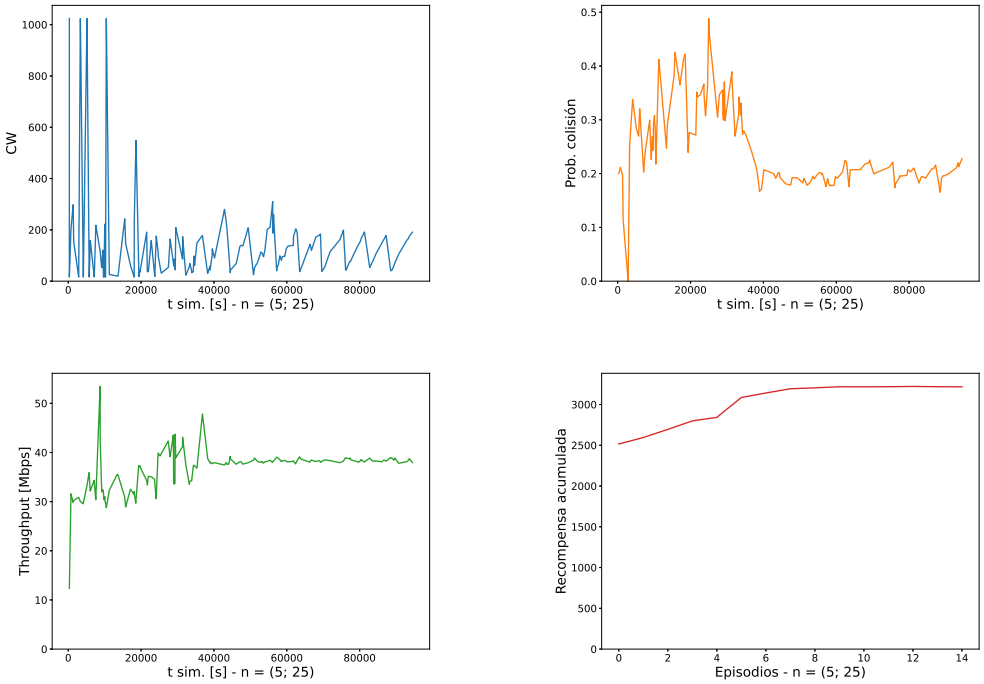

Figura 5.3: Desempeño del método E-CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario dinámico con $n = (5; 25)$ STAs.

5.2.2. Caso TCP

Como se mencionó, es de interés que el método E-CCOD pueda funcionar de manera satisfactoria en escenarios ampliamente utilizados en la realidad actual. Un escenario de operación típico es una red, cuyas STAs y AP intercambian segmentos TCP. Se modificó entonces el código para generar envíos TCP, en sentido UL, DL y UL+DL; además se pasó a estimar la probabilidad de colisión con la siguiente fórmula, tomada de [95], la cual es una aproximación de primer orden al cálculo original (el mismo no puede utilizarse debido a que, en transmisiones TCP, el tráfico DL siempre está presente, por datos o reconocimiento):

$$p = \frac{2rx_{error}}{2rx_{error} + rx_{ok}/2}, \tag{5.1}$$

donde rx_{error} y rx_{ok} corresponden a trazas de capa física de ns-3 [96]. Cabe señalar que el contador de recepciones no satisfactorias (rx_{error}) se multiplica por 2 ya que en una colisión participan (al menos) dos transmisiones; por otro lado, el de recepciones exitosas (rx_{ok}) se divide entre 2, ya que por defecto se consideran las tramas de datos y de reconocimiento, y en este caso solo deben contabilizarse las de datos.

En el gráfico de la Figura 5.5 se muestra la comparación entre la probabilidad

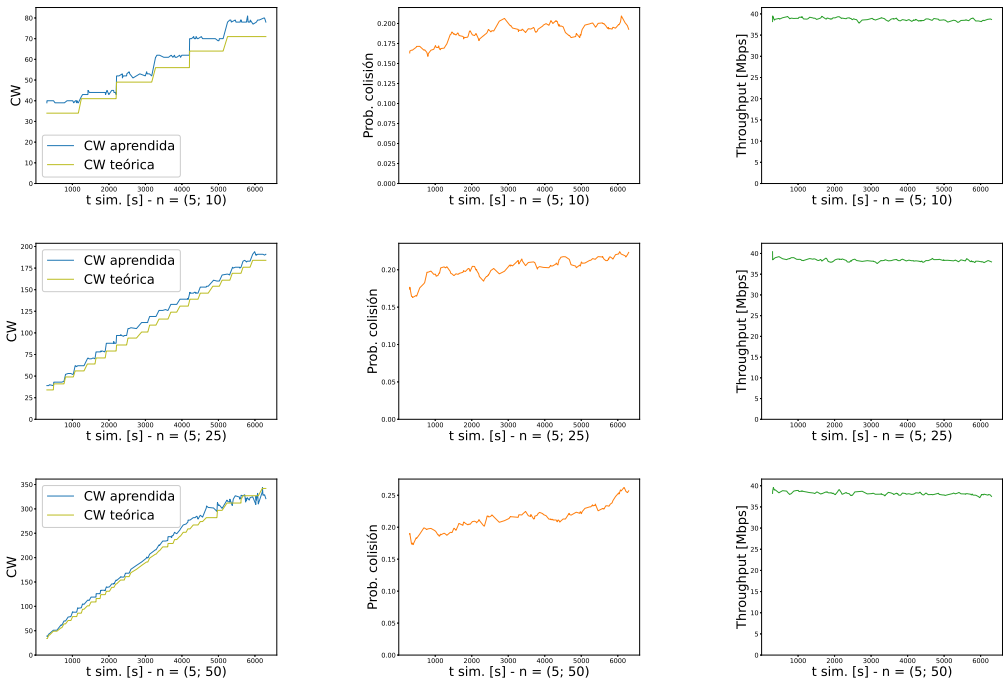

Figura 5.4: Desempeño del método E-CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión y throughput en fase de evaluación, utilizando escenarios dinámicos. Se emplearon distintas cantidades de STAs: $n = (5; 10)$ - fila 1, $n = (5; 25)$ - fila 2 y $n = (5; 50)$ - fila 3. Se aprecia que el algoritmo acompaña la evolución de la red y logra seleccionar una CW cercana a la óptima, incluso cuando se utilizan cantidades de STAs no usadas en la fase de entrenamiento (aunque en este caso lo hace de manera más ruidosa).

de colisión obtenida con la aproximación de primer orden (5.1), con la fórmula implementada originalmente (4.1) y con la expresión analítica (2.2). Dicha comparación se realizó con envíos UDP saturados, ya que es el escenario en que son válidas las tres expresiones. De la misma puede afirmarse que la aproximación funciona muy bien, por lo que es correcto emplearla para la extensión del sistema a operar con redes TCP.

Por otro lado, para poder determinar si el agente logra seleccionar valores de CW cercanos a los óptimos, es necesario conocer estos últimos. Dado que no existen ecuaciones que modelen un escenario con envíos TCP (como sí ocurre con UDP saturado), para determinarlos se realizaron barridos de simulaciones con distintos valores de CW y se seleccionó como óptimo aquel que conseguía el valor más alto de throughput. La Tabla 5.1 presenta algunos valores para ejemplificar este proceso para $n = 10, 25$ (cantidades utilizadas en las posteriores evaluaciones) y envíos UL, DL y UL+DL; en cada caso (i.e. n y sentido) se encuentra resaltada la fila que logró el mejor resultado de throughput. Es importante destacar que el valor de probabilidad de colisión en sentido DL es muy bajo, debido a que en este tipo de envíos el AP es quien transmite tramas de datos y las STAs solo tramas

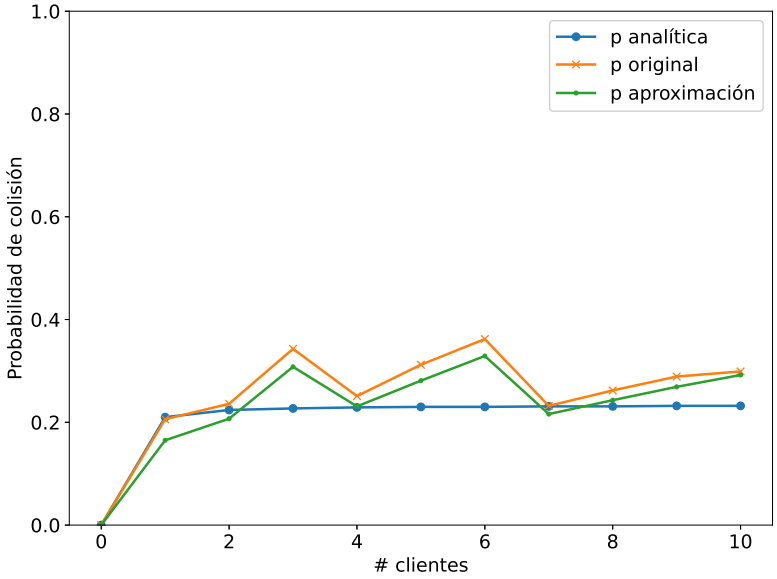

Figura 5.5: Caso UDP saturado. Valores obtenidos mediante modelo de Bianchi y simulaciones en ns-3. Comparación entre expresiones de probabilidad de colisión, utilizando CW óptima para distintas cantidades de STAs. Se aprecia que los resultados obtenidos con la expresión original y la aproximación de primer orden son muy similares entre sí.

con reconocimientos de TCP (muy pequeñas en comparación a las de datos), por lo que ocurren menos colisiones que en el caso UL.

Además, puede observarse que la cantidad de STAs activas en todos los casos está en el entorno de $n' = 5$, más allá de encender $n = 10$ o $n = 25$ STAs en la red y sin importar el sentido de los envíos. Si se analiza el tráfico en sentido DL, se tiene que, dado que se está trabajando con tráfico saturado, el AP siempre tiene una trama de datos para ser enviada; en cambio, las STAs solo envían tramas (con un reconocimiento de TCP) cuando reciben una trama de datos. Si varias STAs de la red tienen tramas para enviar, la probabilidad de que el AP gane el acceso al medio (y potencialmente incremente la cantidad de STAs activas) es muy baja, por ende, el sistema opera con una cantidad relativamente baja de STAs activas. Por otro lado, con el tráfico en sentido UL ocurre algo similar, puesto que las STAs pueden enviar tramas de datos solo luego de recibir una trama con un reconocimiento de TCP, y para ello, el AP debe ganar el acceso al medio. Este comportamiento fue estudiado con mayor profundidad en el trabajo [5].

Habiendo modificado el código para operar en este tipo de redes y contando con valores de referencia para la CW óptima, se pasó a ejecutar el entrenamiento del agente empleando un escenario dinámico, compuesto por múltiples episodios, donde en cada episodio se comenzaba con $n = 10$ y, a la mitad del mismo, se incrementaba a $n = 25$, de forma de permitir a TCP alcanzar un estado de régimen y para poder evaluar luego dos escenarios bien diferenciados en términos de

Tabla 5.1: Caso TCP saturado. Valores obtenidos mediante simulaciones en ns-3. Valores de CW y parámetros asociados para distintos sentidos de envío y cantidades de STAs. Se resalta en verde la CW con la que se obtuvo el valor de throughput más elevado en cada caso.

Sentido	n	n'	CW	p	$S[Mbps]$
UL	10	6	15	0,219	27,44
UL	**10**	**4**	**60**	**0,130**	**34,56**
UL	10	4	120	0,064	30,27
UL	25	7	15	0,235	26,92
UL	**25**	**5**	**60**	**0,151**	**34,56**
UL	25	5	120	0,086	31,63
DL	**10**	**5**	**15**	**0,054**	**29,03**
DL	10	5	20	0,037	27,70
DL	10	4	40	0,021	22,69
DL	25	5	15	0,057	29,05
DL	**25**	**2**	**20**	**0,039**	**29,76**
DL	25	4	40	0,022	22,66
UL+DL	10	2	15	0,099	29,28
UL+DL	**10**	**4**	**50**	**0,113**	**31,46**
UL+DL	10	2	100	0,022	20,14
UL+DL	25	2	15	0,104	29,26
UL+DL	**25**	**5**	**50**	**0,169**	**31,19**
UL+DL	25	2	100	0,023	19,96

cantidad de STAs. La evaluación se realizó empleando dos escenarios estáticos con $n = 10$ y $n = 25$. Se probaron envíos UL, DL y UL+DL. La Tabla 5.2 resume los resultados obtenidos; se omiten los gráficos en este caso, debido a la poca variación de los parámetros. A partir de dichos resultados se puede concluir que el agente logra seleccionar valores de CW cercanos al óptimo (ver Tabla 5.1) para tráfico TCP en sentido UL y DL. Para envíos UL+DL, el agente elige valores de CW más pequeños a los encontrados en el barrido de la Tabla 5.1; no obstante, los valores de throughput y probabilidad de colisión alcanzados con ambos valores de CW (óptimo y aprendido) son similares entre sí.

Se entiende que la manera en que se realizó el entrenamiento no introduce pérdidas significativas a la evaluación del sistema, ya que, como se observó anteriormente, la cantidad de STAs activas, n', se mantiene más o menos constante en un valor relativamente bajo, conforme aumenta la cantidad de STAs, n, en la red, por tratarse de envíos TCP. En cualquier caso, si se requiriera incorporar un caso intermedio (e.g. $n = 18$), bastaría con incluirlo en la configuración del escenario de prueba e incrementar el tiempo de simulación de forma de que cada red (cada valor de n) opere un tiempo suficiente para que se alcance la convergencia de TCP.

Tabla 5.2: Desempeño del método E-CCOD para el caso TCP saturado. Valores obtenidos mediante simulaciones en ns-3. CW aprendida y probabilidad de colisión y throughput registrados en fase de evaluación, utilizando dos escenarios estáticos con $n = 10$ y $n = 25$. Se probaron envíos UL, DL y UL+DL; para los primeros dos casos el algoritmo logró seleccionar una CW cercana a la óptima, mientras que para el último caso, seleccionó una CW levemente más baja.

Sentido	n	n'	CW	p	$S[Mbps]$
UL	10	(2; 8)	(27; 45)	(0,143; 0,220)	33,60
UL	25	(2; 9)	(28; 47)	(0,146; 0,240)	32,96
DL	10	(3; 8)	15	(0,045; 0,074)	28,94
DL	25	(3; 8)	15	(0,048; 0,078)	28,96
UL+DL	10	2	15	(0,089; 0,107)	29,32
UL+DL	25	2	15	(0,090; 0,116)	29,32

5.2.3. Caso con Variaciones en el Tráfico Cursado

Hasta el momento se pudo constatar el correcto funcionamiento del método E-CCOD en redes donde se cursa únicamente tráfico UDP o tráfico TCP, y las variaciones introducidas corresponden a la cantidad de clientes. Ahora se desea plantear un escenario de prueba donde el agente deba reaccionar a cambios introducidos a nivel del tráfico cursado; de las partes anteriores se conoce que los valores óptimos de CW para UDP (ver Tabla 2.2) suelen ser sensiblemente más elevados que los empleados para TCP (ver Tabla 5.1), entonces se busca que el agente pueda adaptarse de buena manera a una red donde en un determinado momento se intercambian segmentos TCP y luego se pasan a intercambiar segmentos UDP.

En concreto, se diseñó un escenario dinámico, compuesto por múltiples episodios, donde en la primera mitad de cada episodio se cursó tráfico TCP en sentido DL y en la segunda mitad se pasó a utilizar tráfico UDP en sentido UL, la cantidad de STAs se mantuvo constante en $n = 25$ por simplicidad. Dicho escenario fue utilizado para realizar el entrenamiento del sistema. En la Figura 5.6 se muestra la evolución de los principales parámetros durante esta fase. Por su parte, para la evaluación se emplearon dos escenarios estáticos: TCP en sentido DL y UDP en sentido UL, ambos con $n = 25$. En la Tabla 5.3 se presentan los valores registrados, de la cual se puede afirmar que el agente, habiendo sido entrenado una sola vez en una red con cambios en el tipo de tráfico cursado, logra funcionar satisfactoriamente en dos redes bien diferenciadas en términos del valor de CW a configurar para maximizar el throughput.

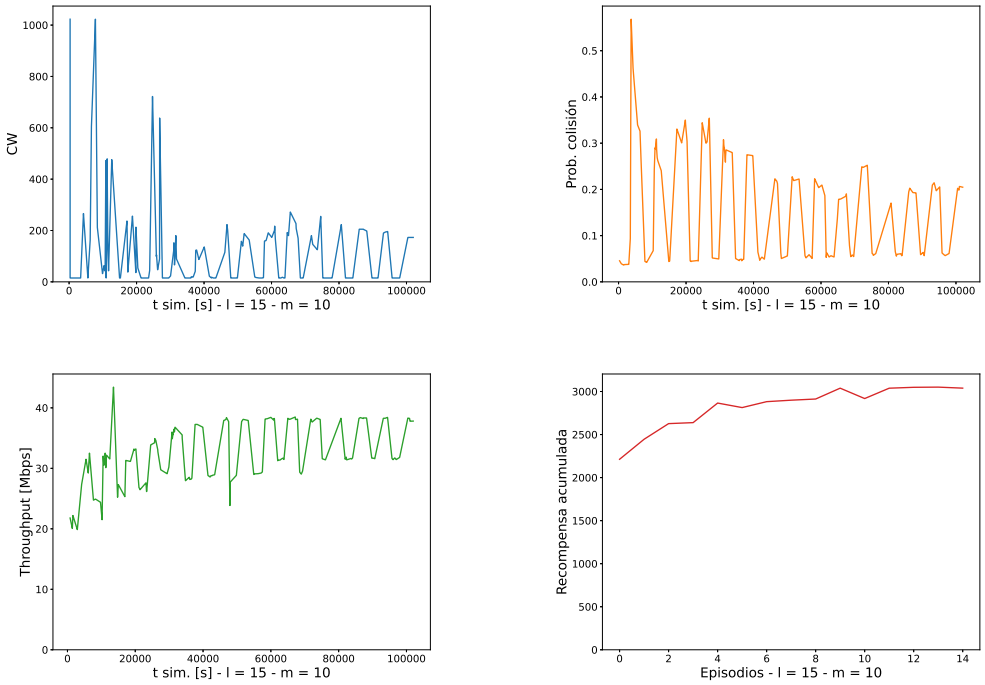

Figura 5.6: Desempeño del método E-CCOD para el caso con variaciones en el tráfico cursado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario dinámico donde en la primera mitad de cada episodio se cursó tráfico TCP DL y en la segunda mitad tráfico UDP UL con $n = 25$ STAs.

Tabla 5.3: Desempeño del método E-CCOD para el caso con variaciones en el tráfico cursado. Valores obtenidos mediante simulaciones en ns-3. CW aprendida y probabilidad de colisión y throughput registrados en fase de evaluación, utilizando dos escenarios estáticos con tráfico TCP DL y UDP UL con $n = 25$ STAs. Se aprecia que el algoritmo logra seleccionar una CW cercana a la óptima en ambos casos.

Tráfico	n	n'	CW	p	$S[Mbps]$
TCP DL	25	2	15	(0,048; 0,067)	31,33
UDP UL	25	23	(165; 183)	(0,183; 0,210)	38,27

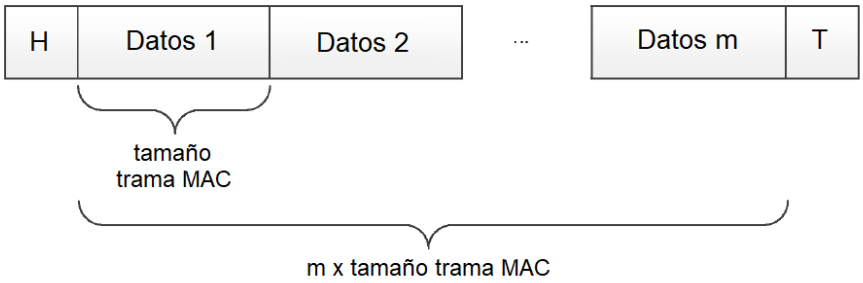

Figura 5.7: Formato de trama agregada utilizada para emular envíos MU en ns-3. Cada bloque de datos representa la transmisión de un cliente 802.11ax diferente; mientras que H y T corresponden al encabezado y cola de la trama, respectivamente.

5.3. Aplicación del Método a Redes con Clientes IEEE 802.11ax y Legacy

En línea con la aprobación de la nueva enmienda del estándar IEEE 802.11 y la inminente etapa de transición en la que las redes tengan clientes 802.11ax y *legacy* en coexistencia, es de interés evaluar qué desempeño puede lograr el agente de DRL, debiendo configurar ahora dos CW: una para que el AP permita el uso de OFDMA con clientes de la nueva enmienda y otra para que los de las enmiendas anteriores (que no implementan OFDMA) accedan al medio de la forma tradicional, tal como se explicó en la Sección 2.2 y como puede verse en el diagrama de la Figura 2.6. Para ello, fue necesario extender el sistema construido para emplear un entorno con clientes de estos dos tipos y un agente que, a partir de la observación del entorno, pueda seleccionar el valor de dos CW.

Debido a que los conceptos de OFDMA se comenzaron a incorporar en ns-3 a partir de la versión 34 (liberada el 14 de julio de 2021) [97], algunas versiones posteriores a la versión utilizada en el sistema trabajado (versión 29), fue necesario recurrir a una abstracción para representar los envíos MU provistos por esta tecnología. Se tiene que, en 802.11ax, la trama de capa física para envíos MU lleva un encabezado y múltiples tramas MAC; los clientes envían una o más tramas MAC dentro de los RUs asignados para su transmisión (ver Sección 2.2). De manera análoga, en la simulación, m clientes 802.11ax se implementaron en una sola STA de ns-3, la cual envía tramas agregadas de largo total $m \times$ *tamaño trama MAC*, así dicha STA debe ganar el acceso al medio solo una vez y, al hacerlo, transmite múltiples segmentos UDP o TCP de una vez (cada segmento representa la transmisión de un cliente 802.11ax). Cabe destacar además que en la simulación se deshabilitó la fragmentación de paquetes. El esquema de la Figura 5.7 representa el formato general de estas tramas agregadas. La abstracción implementada solo puede funcionar en sentido UL, ya que, la funcionalidad de agregación se habilita o deshabilita por equipo, entonces, si se habilitara en el AP para representar los envíos DL de 802.11ax, quedaría habilitada para los envíos a todas las STAs de la red (tanto para las de 802.11ax como para las *legacy*).

Luego de las modificaciones introducidas al entorno para emular la coexistencia de clientes 802.11ax y de versiones anteriores del estándar, se revisaron y extendieron los elementos del sistema de DRL presentados en la Sección 4.1. A continuación se repasan los distintos elementos, modificados adecuadamente para este caso de estudio:

- Al *agente* se lo mantiene en el AP, para continuar aprovechando la visión global que tiene este equipo sobre la red y su capacidad de procesamiento.
- El *estado* del entorno se describe en términos de la media y varianza de múltiples muestras de probabilidad de colisión de la red, calculada a partir de la expresión (5.1), y de la cantidad de STAs activas *legacy*, l', y 802.11ax, m' (de manera análoga a los casos anteriores, diferenciando ahora entre los dos tipos de STAs).
- La *acción* consiste en seleccionar dos valores de CW: CW_{ax} usado para dar lugar a los envíos MU de los clientes de la nueva enmienda (configurado en la STA que envía tramas agregadas) y CW_{legacy} usado para que los de las enmiendas anteriores accedan al medio (configurado en las demás STAs y en el AP). Al igual que antes, el agente no elige directamente valores de CW, sino de un parámetro α que toma valores continuos en $[0, 6]$ y luego calcula la CW usando la expresión (4.2).
- La recompensa usada sigue siendo el throughput de la red, normalizado adecuadamente para que tome valores en $[0, 1]$.

Con el objetivo de obtener resultados preliminares en este escenario de coexistencia, se trabajó con envíos UDP UL, en un escenario estático con cantidades fijas de STAs 802.11ax (representadas con la abstracción comentada) y *legacy*, notadas como m y l, respectivamente. En primer lugar, se realizaron corridas sin intervención del agente y con valores de CW establecidos a mano para obtener valores óptimos de referencia (análogamente a como se hizo con redes TCP). Dado que cuando la STA que emula 802.11ax logra transmitir envía m segmentos, mientras que cuando lo hace una *legacy* envía solo 1, es razonable esperar que los valores de CW_{ax} tiendan a ser pequeños y los de CW_{legacy} tiendan a ser grandes. Para reducir el barrido en simulaciones realizado, se mantuvo fijo el valor de CW_{legacy} en el más grande que se puede configurar y se varió CW_{ax} desde el más pequeño hasta algunos valores más grandes. En la Tabla 5.4 se muestran algunos resultados de esta prueba, resaltándose el caso en que se logró el mejor resultado de throughput. Además, para dicho caso, se registró el reparto de throughput entre dispositivos *legacy* y 802.11ax, donde se obtuvo un valor de 2 Mbps para los primeros y uno de 92 Mbps para los segundos, aproximadamente.

Como siguiente paso, se realizó el entrenamiento del sistema, obteniéndose los resultados que se muestran en la Figura 5.8. La primera observación a realizar es respecto a los valores de throughput alcanzados, los cuales son muy superiores a los vistos en los otros casos, así como los valores de probabilidad de colisión que son sensiblemente bajos comparados con otros casos de envíos UL. Esto sin dudas

Tabla 5.4: Caso de coexistencia entre clientes 802.11ax y *legacy*. Valores obtenidos mediante simulaciones en ns-3. Valores de CW_{legacy}, CW_{ax} y parámetros asociados para distintas cantidades de STAs *legacy* y 802.11ax. Se resalta en verde el par de valores de CW con el que se obtuvo el valor de throughput más elevado.

l	l'	$m = m'$	CW_{legacy}	CW_{ax}	p	$S[Mbps]$
15	**6**	**10**	**1023**	**15**	**0,008**	**93,78**
15	7	10	1023	30	0,008	87,47
15	9	10	1023	60	0,009	77,52
15	10	10	1023	90	0,009	70,09
15	10	10	1023	120	0,010	64,29
15	10	10	1023	180	0,011	55,85

habla de un mejor desempeño del nuevo acceso al medio introducido en 802.11ax, el cual permite un uso mucho más eficiente del canal con mayor envío de datos y menos colisiones. Por otro lado, poniendo foco en el desempeño del agente en sí, se tiene que para CW_{legacy} se selecciona el valor más grande posible, lo cual está alineado con el razonamiento expuesto en el párrafo anterior. En lo que respecta al valor de CW_{ax}, si bien se elige un valor relativamente pequeño, el agente no es preciso en seleccionar el valor más pequeño posible, el cual habilitaría a lograr el mayor throughput posible (ver Tabla 5.4).

Más allá de los resultados específicos obtenidos, es clara la priorización que se da en el sistema a las transmisiones MU de los clientes 802.11ax, excluyendo así a las transmisiones de los clientes *legacy*. Esto es debido a que, en la fase de entrenamiento, se buscó maximizar el throughput de la red (utilizando este parámetro como la recompensa del agente), sin considerar criterios de justicia entre clientes. La idea de justicia entre clientes de distintos tipos (con velocidades de envío de datos diferentes) en redes IEEE 802.11 ha sido ampliamente abordada en múltiples estudios. Algunos de ellos se enfocan en la equidad de tiempo de aire, que se traduce en un reparto justo de ancho de banda del canal a lo largo del tiempo. Existen variaciones en su implementación, como el regulador de token-bucket [98], el planificador de turnos rotativos de déficit de tiempo aire [99], el controlador de ventana de contención [100] y el 802.11e TXOP [101]. Otros trabajos, en cambio, emplean la noción de equidad proporcional para permitir un equilibrio entre la equidad y la eficiencia del espectro. Básicamente, este concepto refiere a que el throughput individual de cada nodo es proporcional a su velocidad de envío de datos. Por ejemplo, en el trabajo [102] se introduce un algoritmo CSMA modificado que implementa esta idea y compara su desempeño con otros criterios de equidad, incluido el de equidad de tiempo de aire.

Teniendo en cuenta lo anterior, se entiende que una recompensa más adecuada sería una que garantice que una proporción del ancho de banda total del canal sea utilizado por clientes de las versiones anteriores del estándar, mientras que el resto pueda ser empleado por los de la nueva versión. La definición de estas proporciones es un desafío en sí mismo, ya que depende de la cantidad de clientes de las distintas

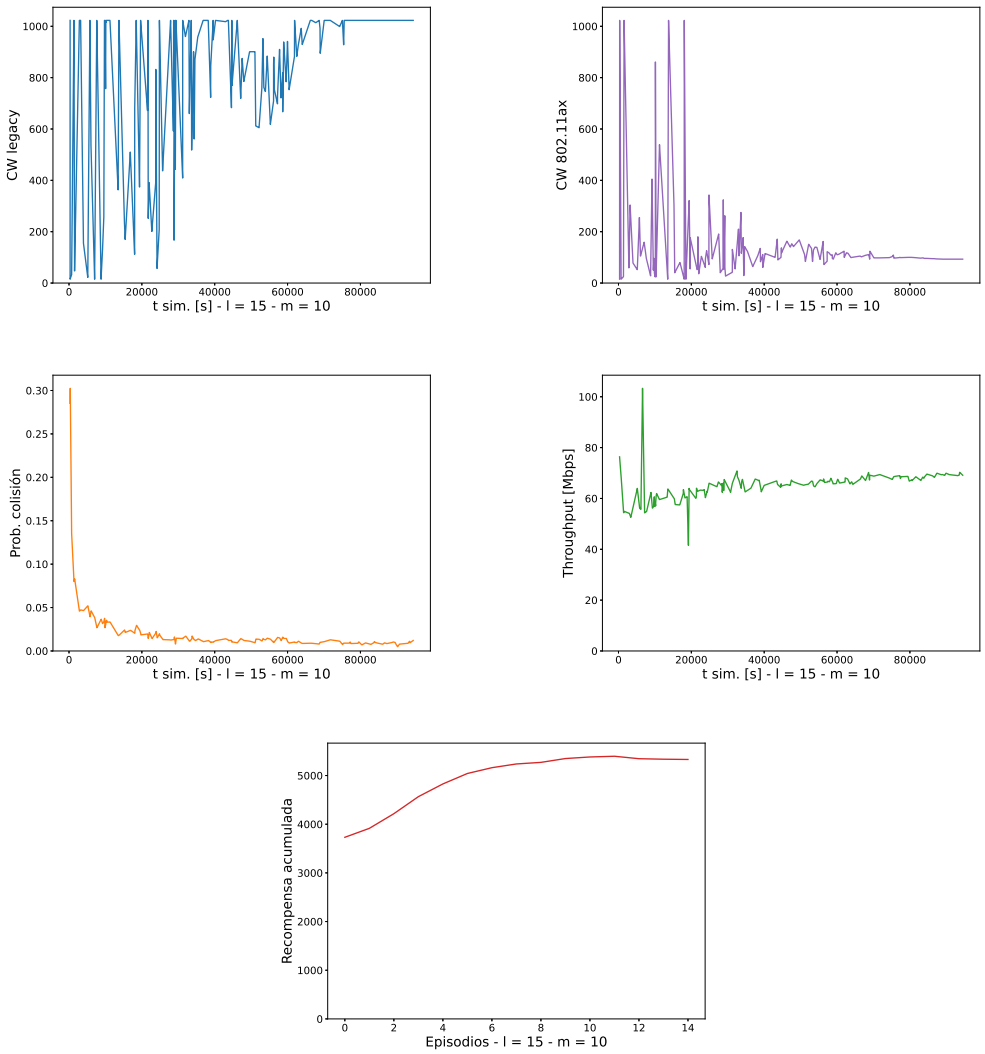

Figura 5.8: Desempeño del método E-CCOD para el caso de coexistencia entre clientes 802.11ax y *legacy*. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW_{legacy}, CW_{ax}, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario estático con $n = 25$ STAs.

versiones que se esperen en la red y de políticas que se quieran implementar en dicha red (podría optarse por darle cierta prioridad a los clientes 802.11ax o, por el contrario, podría buscarse garantizar un buen desempeño de los clientes tradicionales). Quizás, la forma correcta de contemplar estas características, que además varían caso a caso, sea permitir al administrador de red regular la proporción de throughput total "dedicado" a clientes *legacy* y a 802.11ax. Para ello, bastaría con trabajar con una recompensa parametrizable y con reentrenar el agente luego de realizar ajustes en dichas proporciones.

En el presente capítulo se propuso el método E-CCOD para la optimización del rendimiento de redes IEEE 802.11 mediante la predicción correcta de los valores de CW. Luego, se evaluó su desempeño en distintos escenarios de operación realistas (envíos UDP y TCP, sentidos UL, DL y UL+DL, y variaciones en la cantidad de clientes y en el tráfico cursado), pudiendo constatar que dicho método funciona de manera satisfactoria en todos ellos. A continuación, se modificó el sistema construido para poner a prueba un ejemplo de aplicación de una red con clientes 802.11ax y *legacy* en coexistencia. Este tipo de escenarios tiene la particularidad de que el agente debe seleccionar dos valores de CW, uno para cada tipo de clientes. Como resultado se vio que el agente logró elegir el valor más elevado posible de CW para clientes 802.11ax y un valor bajo de CW para clientes de versiones anteriores (pese a no haber elegido el más bajo posible), lo cual es consistente con el objetivo de maximizar el throughput de la red, pero evidencia el problema de un reparto justo de recursos entre estos dos tipos de clientes. Finalmente, se discutió sobre conceptos de equidad de tiempo de aire y de equidad proporcional, y se propuso una modificación al agente para pasar a emplear una recompensa parametrizable, la cual garantice que determinada proporción del throughput total sea utilizado por clientes de versiones anteriores y el restante por clientes de la nueva versión.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

El foco principal del presente trabajo fue estudiar el potencial de la aplicación de DRL a la optimización del control de acceso al medio en redes IEEE 802.11. Dicha optimización se implementó mediante un algoritmo de ML que selecciona valores adecuados de CW según la cantidad de clientes activos, el tipo de clientes (802.11ax o *legacy*) y el tipo de tráfico cursado (TCP o UDP y distintos sentidos de envío). Adicionalmente, se generó conocimiento sobre la nueva enmienda de IEEE 802.11, sus objetivos, las principales funcionalidades que se introducen o se extienden de enmiendas anteriores y su desempeño en escenarios educativos (en base a simulaciones y a pruebas con equipos comerciales).

En las evaluaciones con equipos comerciales se trabajó con redes de dispositivos 802.11ax solamente, así como también con redes compuestas por dispositivos de la nueva versión del estándar y de versiones anteriores en coexistencia. Las pruebas exhaustivas realizadas revelaron una inmadurez en las implementaciones de 802.11ax por el momento. Por un lado, se detectó el problema con clientes Ubuntu y el AP Aruba, donde luego de algunos clientes en la celda las conexiones se volvían inestables y los mismos se desasociaban. Por otro lado, los resultados con el AP Cisco Catalyst no fueron satisfactorios, siendo peores que los registrados con 802.11n. Si bien en las pruebas con uno o dos clientes (pruebas básicas) se habían visto mejoras en el rendimiento de Wi-Fi 6 sobre Wi-Fi 5 en algunos casos, a medida que la red comienza a escalar, el rendimiento cae significativamente. No obstante, es de esperar que estos problemas se resuelvan a medida que la tecnología madure. Llama la atención la preponderancia del tráfico UL sobre el DL visto en las pruebas, algo que sucedió de manera inversa en la prueba de referencia con 802.11n. Dado que el nuevo estándar no prioriza explícitamente el tráfico UL, lo que se observa puede deberse a que en ciertas implementaciones del proceso encargado de la asignación de recursos (*scheduler* de RUs) se busque favorecer las transmisiones en ese sentido para responder a nuevas características del estándar, como los escenarios de IoT.

En cuanto a los simuladores de red, también existe una inmadurez en la imple-

mentación del nuevo estándar, lo que se traduce en limitaciones en las funcionalidades ofrecidas. En los dos simuladores de red empleados, al momento de realizar las pruebas no se había implementado fielmente el mecanismo de acceso al medio propuesto en 802.11ax, lo que compromete los resultados obtenidos. Sin embargo, a partir de la versión 34 (liberada el 14 de julio de 2021) se comienzan a incorporar conceptos de OFDMA, lo que habilita a conducir pruebas más representativas a futuro.

Dejando de lado los comentarios asociados a la evaluación de desempeño del nuevo estándar y la situación actual de sus implementaciones, es de interés poner foco en el trabajo realizado en torno a la aplicación de DRL a la optimización del control de acceso al medio en redes IEEE 802.11. En primer lugar, se analizó un modelo teórico de este tipo de redes (el modelo de Bianchi) para comprender, al menos en un caso ideal, los principales parámetros que deben considerarse para la búsqueda de una CW tal que se maximice el throughput de la red. De este análisis se puede afirmar que la probabilidad de colisión y la cantidad de STAs son dos parámetros de suma importancia.

Posteriormente, se realizó una revisión bibliográfica para comprender si DRL era una técnica de ML adecuada para abordar la problemática a trabajar y para conocer el trabajo relacionado. De esta forma se encontró un antecedente del uso de DRL para la optimización del control de acceso, el cual tenía problemas serios de diseño e implementación que conducían a un desempeño inaceptable en la mayoría de los casos probados. El agente de DRL propuesto en dicho trabajo solo funcionaba de manera razonable si se lo entrenaba con una cantidad fija de STAs y luego se lo evaluaba con la misma cantidad, lo cual no aporta valor en la práctica y está alejado de lo que es la operación real de una red. No obstante, este estudio fue de gran utilidad como primer acercamiento a la temática y por brindar una implementación que sirvió como punto de partida para el sistema construido.

Entonces, tomando como base el trabajo previo mencionado, y a partir del análisis del modelo teórico de Bianchi, se propuso el método E-CCOD, el cual incorpora como variables de estado la probabilidad de colisión y la cantidad de STAs activas en la red. De esta forma se obtuvo un desempeño satisfactorio del agente de DRL en redes donde se varió la cantidad de STAs significativamente (la máxima variación probada fue de 5 a 50). Luego, se extendió su funcionamiento a escenarios típicos de la realidad actual: envíos TCP en sentido UL, DL y UL+DL y redes que en distintos momentos cursan distinto tipo de tráfico (se probaron dos tipos bien diferenciados en términos del valor de CW óptimo a elegir: TCP DL y UDP UL), logrando resultados exitosos en todos ellos. Finalmente, se implementó un ejemplo de operación en redes donde coexisten clientes 802.11ax y *legacy*, debiendo una vez más extender al algoritmo, en este caso para que fuera capaz de seleccionar dos valores de CW.

Los resultados del último experimento muestran una mejora significativa en los valores de throughput y de probabilidad de colisión, en comparación a los demás escenarios evaluados con el mismo tipo de tráfico (UDP UL). En concreto, este fue el único experimento cuyos valores de CW no estuvieron muy cercanos a los óptimos, aunque sí fue clara la preferencia a configurar el valor de CW más grande

posible para los clientes *legacy* y valores pequeños para los 802.11ax. Más allá de eso, dichos resultados indican que, efectivamente, el nuevo acceso al medio previsto por 802.11ax logra un uso mucho más eficiente del canal. Sin embargo, también evidencia que, si no se configuran criterios de justicia entre clientes 802.11ax y *legacy* de manera explícita, la tendencia natural de un sistema cuyo único objetivo es maximizar el throughput de la red es la de priorizar las transmisiones multiusuario de los clientes del primer tipo, frente a las transmisiones (de un solo usuario) de los clientes del segundo tipo. Por esta razón, en el presente trabajo se discutieron alternativas para modificar el sistema construido en pro de introducir estos criterios, como por ejemplo emplear una recompensa parametrizable, la cual garantice que determinada proporción del throughput total sea utilizado por clientes de versiones anteriores y el restante por clientes de la nueva versión.

6.2. Trabajo Futuro

Como trabajo futuro se plantea repetir las medidas con equipos comerciales una vez se hayan liberado versiones de software de los equipamientos de red y de los dispositivos terminales más estables. En relación a esto, sería de gran interés para la academia, la industria en general y, en particular, para Ceibal incorporar a esta evaluación una mayor cantidad de modelos de equipos y nuevos escenarios relacionados a la educación que no sean necesariamente un salón de clases (e.g. salón de conferencias, patio exterior, etc.). De la misma forma, se podría aprovechar que se encuentra disponible la versión 34 o superior de ns-3 para implementar nuevamente las simulaciones realizadas. Esta versión promete haber incorporado conceptos de OFDMA, por lo que su representación del nuevo acceso al medio debería ser más fidedigna.

En lo que refiere al desarrollo y evaluación de desempeño del método E-CCOD, si bien se probaron distintas configuraciones de red, sería bueno complementarlas incorporando nuevos casos de uso o bien profundizando en los casos de uso ya planteados con pruebas más exhaustivas. En particular, se debe seguir trabajando en el escenario de coexistencia de clientes 802.11ax y *legacy*. En primer lugar, se debe realizar una implementación del sistema en una versión de ns-3 más reciente, de forma de poder generar los envíos multiusuario de OFDMA con funcionalidades del simulador en lugar de hacerlo con una abstracción. Para esto, es necesario además utilizar la versión 1.0.1 o superior de la herramienta ns3-gym, la cual permite trabajar con versiones de ns-3 superiores a 29. Luego, sería muy interesante evaluar cómo reacciona el algoritmo frente a cambios en la cantidad de clientes o en el tipo de tráfico en una red de este estilo. Finalmente, es de suma relevancia extender la discusión realizada respecto de emplear una recompensa que incorpore criterios de justicia entre los clientes de diferentes versiones del estándar. Las distintas recompensas consideradas deberían ponerse a prueba a nivel de simulaciones para comprender cómo impactan en el desempeño del agente y, en consecuencia, de la red.

Por otro lado, si se quisiera implementar el método construido para optimizar el control de acceso en redes reales (e.g. las de Ceibal), se debe definir de qué manera

y con qué frecuencia se entrenará el modelo, para luego cargarlo en los equipos de red físicos. Por supuesto, estos equipos deberán ser de código abierto, ya que los equipos comerciales, al trabajar con software privado, no permiten adaptaciones o agregados. En el presente trabajo se puso foco en una sola técnica de DRL (DDPG); sin embargo el trabajo tomado de base planteaba dos (DDPG y DQN) y existen otras. Previo a llevar este sistema a la práctica sería bueno evaluar otras alternativas y seleccionar la más adecuada en términos de precisión, rapidez de convergencia, consumo de recursos, etc.

Como última línea de acción a futuro planteada, podría trabajarse en extender la operación del método E-CCOD a redes con múltiples APs. En ese caso, en cada AP se ubicaría un agente de DRL, los cuales intercambiarían información útil (a determinar) entre sí y, de esta manera, podrían incorporar información del estado de las celdas vecinas a la selección de los valores de CW.

Referencias

[1] Simon Kemp. Digital 2022: Global Overview Report. https://datareportal.com/reports/digital-2022-global-overview-report, Ene 2022. Accedido en 05-2022.

[2] Ceibal. https://www.ceibal.edu.uy/es. Accedido en 05-2022.

[3] Ceibal en cifras. https://www.ceibal.edu.uy/es/articulo/ceibal-en-cifras. Accedido en 05-2022.

[4] CREA se consolida como plataforma educativa. https://www.ceibal.edu.uy/es/articulo/crea-se-consolida-como-plataforma-educativa, Oct 2021. Accedido en 05-2022.

[5] Federico Larroca and Fernanda Rodríguez. An Overview of WLAN Performance, Some Important Case-Scenarios and Their Associated Models. *Wirel. Pers. Commun.*, 79(1):131–184, Nov 2014.

[6] IEEE 802.11ax-2021 - IEEE Standard for Information Technology. https://standards.ieee.org/standard/802_11ax-2021.html. Accedido en 05-2022.

[7] Evgeny Khorov, Anton Kiryanov, Andrey Lyakhov, and Giuseppe Bianchi. A Tutorial on IEEE 802.11ax High Efficiency WLANs. *IEEE Communications Surveys Tutorials*, 21(1):197–216, 2019.

[8] Pierluigi Gallo, Katarzyna Kosek-Szott, Szymon Szott, and Ilenia Tinnirello. CADWAN: A Control Architecture for Dense WiFi Access Networks. *IEEE Communications Magazine*, 56(1):194–201, 2018.

[9] 10 Awesome Machine Learning Applications of Today. https://www.projectpro.io/article/10-awesome-machine-learning-applications-of-today/364, Mar 2022. Accedido en 05-2022.

[10] Nikita Butakov, Loren Jan, Wilson, Wenting Sun, and Angel Barranco. Machine learning use cases: how to design ML architectures for today's telecom systems. https://www.ericsson.com/en/blog/2021/5/machine-learning-use-cases-in-telecom, May 2021. Accedido en 05-2022.

[11] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[12] Tarun Joshi, Anindo Mukherjee, Younghwan Yoo, and Dharma P. Agrawal. Airtime Fairness for IEEE 802.11 Multirate Networks. *IEEE Transactions on Mobile Computing*, 7(4):513–527, 2008.

[13] Fabián Frommel. Rlinwifi. https://github.com/ffrommel/RLinWiFi, 2021. Accedido en 05-2022.

[14] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[15] Der-Jiunn Deng, Chih-Heng Ke, Hsiao-Hwa Chen, and Yueh-Min Huang. Contention window optimization for ieee 802.11 DCF access control. *IEEE Transactions on Wireless Communications*, 7(12):5129–5135, 2008.

[16] Kunho Hong, SuKyoung Lee, Kyungsoo Kim, and YoonHyuk Kim. Channel condition based contention window adaptation in ieee 802.11 wlans. *IEEE Transactions on Communications*, 60(2):469–478, 2012.

[17] Witold Wydmański and Szymon Szott. Contention Window Optimization in IEEE 802.11ax Networks with Deep Reinforcement Learning. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2021.

[18] What is a Working Group? https://standards.ieee.org/develop/mobilizing-working-group/wg/. Accedido en 05-2022.

[19] Industrial Scientific and Medical (ISM) Bands. http://www.wirelesscommunication.nl/reference/chaptr01/dtmmsyst/ism.htm. Accedido en 05-2022.

[20] Eustathia Ziouva and Theodore Antonakopoulos. Csma/ca performance under high traffic conditions: throughput and delay analysis. *Computer Communications*, 25(3):313–321, 2002.

[21] Saad Ayoob and Abudul Jabbar. Modeling a multi-hop ad-hoc network using chain and cross-topologies. pages 135–140, Dic 2013.

[22] Wi-Fi Alliance. https://www.wi-fi.org/. Accedido en 05-2022.

[23] Moh Chuan Tan, Minghui Li, Qammer H. Abbasi, and Muhammad Ali Imran. A Wideband Beamforming Antenna Array for 802.11ac and 4.9 GHz in Modern Transportation Market. *IEEE Transactions on Vehicular Technology*, 69(3):2659–2670, 2020.

[24] Official IEEE 802.11 Working Group Project Timelines. https://www.ieee802.org/11/Reports/802.11_Timelines.htm. Accedido en 05-2022.

[25] Status of Project IEEE 802.11ax. https://www.ieee802.org/11/Reports/tgax_update.htm. Accedido en 05-2022.

[26] François Vergès. MCS Table (Updated with 802.11ax Data Rates). https://www.semfionetworks.com/blog/mcs-table-updated-with-80211ax-data-rates. Accedido en 05-2022.

[27] Aruba. 802.11ax (whitepaper). https://www.arubanetworks.com/assets/wp/WP_802.11AX.pdf.

[28] Eve Danel. Wi-Fi 6's OFDMA Challenges Make Verification Crucial. https://www.rfglobalnet.com/doc/wi-fi-s-ofdma-challenges-make-verification-crucial-0001. Accedido en 05-2022.

[29] O. Cabral, A. Segarra, and F. Velez. Implementation of Multi-service IEEE 802.11e Block Acknowledgement Policies. *IAENG International Journal of Computer Science*, 36, Ene 2008.

[30] Eldad Perahia. High Efficiency Wi-Fi: 802.11ax. https://www.ekahau.com/wp-content/uploads/2020/06/Webinar-slides-802.11ax-Sneak-Peek-%E2%80%93-The-Next-Generation-Wi-Fi.pdf, Mar 2017.

[31] François Baccelli and Serguei Foss. On the saturation rule for the stability of queues. *Journal of Applied Probability*, 32(2):494–507, 1995.

[32] Anurag Kumar and Deepak Patil. Stability and Throughput Analysis of Unslotted CDMA-ALOHA with Finite Number of Users and Code Sharing. *Telecommun. Syst.*, 8(2–4):257–275, Dic 1997.

[33] B.P. Crow, I. Widjaja, J.G. Kim, and P. Sakai. Investigation of the IEEE 802.11 medium access control (MAC) sublayer functions. In *Proceedings of INFOCOM '97*, volume 1, pages 126–133 vol.1, 1997.

[34] Tien-Shin Ho and Kwang-Cheng Chen. Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol. In *Proceedings of PIMRC '96 - 7th International Symposium on Personal, Indoor, and Mobile Communications*, volume 2, pages 407–411 vol.2, 1996.

[35] F. Cali, M. Conti, and E. Gregori. IEEE 802.11 wireless LAN: capacity analysis and protocol enhancement. In *Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98*, volume 1, pages 142–149 vol.1, 1998.

[36] Task Group AX. TGax Simulation Scenarios, Nov 2015.

[37] Brandon Butler. How Google is speeding up the Internet. https://www.networkworld.com/article/3218084/how-google-is-speeding-up-the-internet.html, Ago 2017. Accedido en 05-2022.

[38] Carlo Grazia and Natale Patriciello. TCP small queues and WiFi aggregation. https://lwn.net/Articles/757643/, Jun 2018. Accedido en 05-2022.

[39] ns-3. https://www.nsnam.org/. Accedido en 05-2022.

[40] Komondor. https://github.com/wn-upf/Komondor. Accedido en 05-2022.

[41] ns-3.31. https://www.nsnam.org/releases/ns-3-31/, 2020. Accedido en 05-2022.

[42] Wi-Fi Certified 6. https://www.wi-fi.org/discover-wi-fi/wi-fi-certified-6. Accedido en 05-2022.

[43] Cisco. Cisco Catalyst 9115 Series Wi-Fi 6 Access Points Data Sheet. https://www.cisco.com/c/en/us/products/collateral/wireless/catalyst-9100ax-access-points/datasheet-c78-741988.html. Accedido en 05-2022.

[44] 2.4/5 GHz 4/6 dBi 4 Element Indoor/Outdoor Omni Antenna With RPTNC. http://shop.acceltex.com/product_info.php?cPath=0_21&products_id=46&osCsid=uc3jr1rhrvjb5dntatsoiuplj7. Accedido en 05-2022.

[45] Cisco. MR46 Wi-Fi 6 (802.11ax) with Multigigabit Ethernet. https://meraki.cisco.com/product/wi-fi/indoor-access-points/mr46/. Accedido en 05-2022.

[46] Aruba. Aruba 510 Series Indoor Access Points. https://www.arubanetworks.com/products/wireless/access-points/indoor-access-points/510-series/. Accedido en 05-2022.

[47] Plan Ceibal. Sirio. https://www.ceibal.edu.uy/es/articulo/sirio, Feb 2020. Accedido en 05-2022.

[48] Intel. Intel® Wireless-AC 9461. https://www.intel.la/content/www/xl/es/products/wireless/wireless-products/dual-band-wireless-ac-9461.html. Accedido en 05-2022.

[49] Intel. Intel® Wi-Fi 6 AX200. https://ark.intel.com/content/www/es/es/ark/products/189347/intel-wi-fi-6-ax200-gig.html. Accedido en 05-2022.

[50] Intel. Intel® Dual Band Wireless-N 7260. https://ark.intel.com/content/www/us/en/ark/products/75440/intel-dual-band-wireless-n-7260.html. Accedido en 05-2022.

[51] Germán Capdehourat, Germán Álvarez, Martín Álvarez, Pedro Porteiro, and Fernando Bagalciague. High density emulation platform for Wi-Fi performance testing. *Ad Hoc Networks*, 70:1–13, 2018.

[52] iPerf - The ultimate speed test tool for TCP, UDP and SCTP. https://iperf.fr/. Accedido en 05-2022.

[53] Hongjing Ji, Osama Alfarraj, and Amr Tolba. Artificial intelligence-empowered edge of vehicles: Architecture, enabling technologies, and applications. *IEEE Access*, PP:1–1, Mar 2020.

[54] D.L. Poole and A.K. Mackworth. *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, 2010.

[55] John McCarthy. What is artificial intelligence? http://jmc.stanford.edu/articles/whatisai.html, 2007.

[56] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: from theory to algorithms*. 2014.

[57] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[58] Li Deng and Dong Yu. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[59] Arne Wolfewicz. Deep learning vs. machine learning – What's the difference? https://levity.ai/blog/difference-machine-learning-deep-learning. Accedido en 05-2022.

[60] Funciones de activación sigmoide, tanh, relu. https://programmerclick.com/article/3911149818/. Accedido en 05-2022.

[61] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989.

[62] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[63] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[64] Isaac R. Galatzer-Levy, Kelly V. Ruggles, and Zhe Chen. Data Science in the Research Domain Criteria Era: Relevance of Machine Learning to the Study of Stress Pathology, Recovery, and Resilience. *Chronic Stress*, 2:2470547017747553, 2018. PMID: 29527592.

[65] Christopher Watkins and Peter Dayan. Technical Note: Q-Learning. *Machine Learning*, 8:279–292, May 1992.

[66] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

[67] Jan Peters and Stefan Schaal. Policy Gradient Methods for Robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2219–2225, 2006.

[68] Key Concepts in RL. https://spinningup.openai.com/en/latest/spinningup/rl_intro.html. Accedido en 05-2022.

[69] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J. Glaser. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms. *Journal of Magnetic Resonance*, 172(2):296–305, 2005.

[70] Vijay Konda and John Tsitsiklis. Actor-Critic Algorithms. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[71] J.L. McClelland and D.E. Rumelhart. *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*. Bradford book. MIT Press, 1989.

[72] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 3, pages 2619–2624 Vol.3, 2004.

[73] Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Autonomous Inverted Helicopter Flight via Reinforcement Learning. In Marcelo H. Ang and Oussama Khatib, editors, *Experimental Robotics IX*, pages 363–372, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[74] Michael J. Kearns, Diane J. Litman, Satinder Singh, and Marilyn A. Walker. Optimizing dialogue management with reinforcement learning: Experiments with the njfun system. *CoRR*, abs/1106.0676, 2011.

[75] Gerald Tesauro. Temporal Difference Learning and TD-Gammon. *Commun. ACM*, 38(3):58–68, Mar 1995.

[76] Aske Plaat. Deep Reinforcement Learning. *CoRR*, abs/2201.02135, 2022.

[77] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.

[78] Yi-Han Xu, Cheng-Cheng Yang, Min Hua, and Wen Zhou. Deep Deterministic Policy Gradient (DDPG)-Based Resource Allocation Scheme for NOMA Vehicular Communications. *IEEE Access*, 8:18797–18807, 2020.

[79] Haixia Peng and Xuemin Sherman Shen. DDPG-based Resource Management for MEC/UAV-Assisted Vehicular Networks. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–6, 2020.

[80] Po-Chen Chen, Yen-Chen Chen, Wei-Hsiang Huang, Chih-Wei Huang, and Olav Tirkkonen. DDPG-Based Radio Resource Management for User Interactive Mobile Edge Networks. In *2020 2nd 6G Wireless Summit (6G SUMMIT)*, pages 1–5, 2020.

[81] Yueyue Dai, Ke Zhang, Sabita Maharjan, and Yan Zhang. Edge Intelligence for Energy-Efficient Computation Offloading and Resource Allocation in 5G Beyond. *IEEE Transactions on Vehicular Technology*, 69(10):12175–12186, 2020.

[82] Antonios Tsourdos, Ir. Adhi Dharma Permana, Dewi H. Budiarti, Hyo-Sang Shin, and Chang-Hun Lee. Developing Flight Control Policy Using Deep Deterministic Policy Gradient. In *2019 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*, pages 1–7, 2019.

[83] Kai Li, Yousef Emami, Wei Ni, Eduardo Tovar, and Zhu Han. Onboard Deep Deterministic Policy Gradients for Online Flight Resource Allocation of UAVs. *IEEE Networking Letters*, 2(3):106–110, 2020.

[84] Pablo Serrano, Paul Patras, Andrea Mannocci, Vincenzo Mancuso, and Albert Banchs. Control theoretic optimization of 802.11 WLANs: Implementation and experimental evaluation. *Computer Networks*, 57(1):258–272, 2013.

[85] Mehmet Karaca, Saeed Bastani, and Bjorn Landfeldt. Modifying Backoff Freezing Mechanism to Optimize Dense IEEE 802.11 Networks. *IEEE Transactions on Vehicular Technology*, 66(10):9470–9482, 2017.

[86] Francesc Wilhelmi, Sergio Barrachina-Muñoz, Boris Bellalta, Cristina Cano, Anders Jonsson, and Vishnu Ram. A flexible machine-learning-aware architecture for future WLANs. *IEEE Communications Magazine*, 58(3):25–31, 2020.

[87] Thomas Sandholm, Bernardo Huberman, Belal Hamzeh, and Scott Clearwater. Learning to wait: Wi-fi contention control using load-based predictions. *arXiv preprint arXiv:1912.06747*, 2019.

[88] Jingjing Wang, Chunxiao Jiang, Haijun Zhang, Yong Ren, Kwang-Cheng Chen, and Lajos Hanzo. Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks. *IEEE Communications Surveys Tutorials*, 22(3):1472–1514, 2020.

[89] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Communications Surveys Tutorials*, 21(3):2224–2287, 2019.

[90] ns-3.29. https://www.nsnam.org/releases/ns-3-29/, 2018. Accedido en 05-2022.

[91] python. https://www.python.org/. Accedido en 05-2022.

[92] PyTorch. https://pytorch.org/. Accedido en 05-2022.

[93] NumPy. https://numpy.org/. Accedido en 05-2022.

[94] ns3-gym: OpenAI Gym integration. https://apps.nsnam.org/app/ns3-gym/. Accedido en 05-2022.

[95] A strange finding on collision number. https://groups.google.com/g/ns-3-users/c/7byjPHGIcmo/m/06p-MQIUG8YJ, 2011. Accedido en 05-2022.

[96] The list of all trace sources. https://www.nsnam.org/docs/release/3.18/doxygen/group__trace_source_list.html. Accedido en 05-2022.

[97] ns-3.34. https://www.nsnam.org/releases/ns-3-34/, 2021. Accedido en 05-2022.

[98] Huaizhou SHI, R. Venkatesha Prasad, Ertan Onur, and I.G.M.M. Niemegeers. Fairness in Wireless Networks:Issues, Measures and Challenges. *IEEE Communications Surveys Tutorials*, 16(1):5–24, 2014.

[99] Roberto Riggio, Daniele Miorandi, and Imrich Chlamtac. Airtime Deficit Round Robin (ADRR) packet scheduling algorithm. In *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages 647–652, 2008.

[100] Tarun Joshi, Anindo Mukherjee, Younghwan Yoo, and Dharma P. Agrawal. Airtime Fairness for IEEE 802.11 Multirate Networks. *IEEE Transactions on Mobile Computing*, 7(4):513–527, 2008.

[101] Li Bin Jiang and Soung Chang Liew. Proportional fairness in wireless LANs and ad hoc networks. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 3, pages 1551–1556 Vol. 3, 2005.

[102] Sundaresan Krishnan and Prasanna Chaporkar. Stochastic approximation based on-line algorithm for fairness in multi-rate wireless lans. *Wireless Networks*, 23(5):1563–1574, Jul 2017.

Índice de tablas

2.1. Principales enmiendas del estándar IEEE 802.11. 9

2.2. Caso UDP saturado. Valores obtenidos mediante modelo de Bianchi. *CW* óptima y parámetros asociados para distintas cantidades de STAs. 17

2.3. Parámetros de configuración utilizados para la evaluación de desempeño de IEEE 802.11ax en base a simulaciones y a pruebas con equipos comerciales. 20

4.1. Caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Probabilidad de colisión y throughput registrados cuando se configura *CW* óptima para distintas cantidades de STAs. Se observa que estos valores son razonables y están cercanos a los hallados teóricamente. 45

4.2. Desempeño del método CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. *CW* aprendida y probabilidad de colisión y throughput registrados en fase de evaluación, utilizando escenarios estáticos (Est.) y dinámicos (Din.). Se observa que el algoritmo selecciona una *CW* cercana a la óptima solo cuando se lo entrena con una cantidad fija de STAs y luego se lo evalúa con la misma cantidad (fila 3). 45

5.1. Caso TCP saturado. Valores obtenidos mediante simulaciones en ns-3. Valores de *CW* y parámetros asociados para distintos sentidos de envío y cantidades de STAs. Se resalta en verde la *CW* con la que se obtuvo el valor de throughput más elevado en cada caso. . . 60

5.2. Desempeño del método E-CCOD para el caso TCP saturado. Valores obtenidos mediante simulaciones en ns-3. *CW* aprendida y probabilidad de colisión y throughput registrados en fase de evaluación, utilizando dos escenarios estáticos con $n = 10$ y $n = 25$. Se probaron envíos UL, DL y UL+DL; para los primeros dos casos el algoritmo logró seleccionar una *CW* cercana a la óptima, mientras que para el último caso, seleccionó una *CW* levemente más baja. . 61

5.3. Desempeño del método E-CCOD para el caso con variaciones en el tráfico cursado. Valores obtenidos mediante simulaciones en ns-3. CW aprendida y probabilidad de colisión y throughput registrados en fase de evaluación, utilizando dos escenarios estáticos con tráfico TCP DL y UDP UL con $n = 25$ STAs. Se aprecia que el algoritmo logra seleccionar una CW cercana a la óptima en ambos casos. . . 62
5.4. Caso de coexistencia entre clientes 802.11ax y *legacy*. Valores obtenidos mediante simulaciones en ns-3. Valores de CW_{legacy}, CW_{ax} y parámetros asociados para distintas cantidades de STAs *legacy* y 802.11ax. Se resalta en verde el par de valores de CW con el que se obtuvo el valor de throughput más elevado. 65

Índice de figuras

2.1. Diagrama de tiempos del protocolo CSMA/CA (adaptada de [21]). 8
2.2. Diagramas de constelación para 16-, 64-, 256- y 1024-QAM (adaptada de [27]). 11
2.3. Formato de trama de capa física de 802.11ax (adaptada de [7]). . . 11
2.4. Comparación entre transmisiones UL sobre OFDM y OFDMA. Puede verse que con OFDMA la asignación de recursos es mucho más granular (adaptada de [28]). 12
2.5. Diagrama de tiempos simplificado para transmisiones DL sobre OFDM y OFDMA. Puede verse que con OFDMA el tiempo de aire es utilizado de forma mucho más eficiente (adaptada de [30]). 12
2.6. Diagrama de tiempos del protocolo CSMA/CA adaptado para su operación con OFDMA. En el primer eje se muestran las tramas DL enviadas por el AP para solicitar a las STAs sus transmisiones (TF) y para reconocerlas (BA), así como las tramas UL MU de datos de las STAs 802.11ax. Por otro lado, en el segundo eje se representa una STA que intenta acceder al medio de forma tradicional, la cual lo hace con mucha menor probabilidad que el AP 802.11ax. 13
2.7. Distribuciones de clientes y AP en plano XY utilizadas para la evaluación de desempeño de IEEE 802.11ax en base a simulaciones y a pruebas con equipos comerciales. 19
2.8. Desempeño de IEEE 802.11ax en base a simulaciones. Evolución de throughput para envíos TCP DL y UL+DL, utilizando 802.11n, ac y ax. Gráfico construido con ns-3. Se observa que 802.11ax no destaca sobre las demás versiones probadas. 21
2.9. Desempeño de IEEE 802.11ax en base a simulaciones. Evolución de throughput para envíos UDP DL, utilizando 802.11ax. Gráfico construido con ns-3 y Komondor. Se observa que con ns-3 se produce una ligera disminución del throughput conforme se incrementan los clientes en la red, algo que no ocurre con Komondor. 22
2.10. Diagrama de maqueta de red montada para la evaluación de desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. 22
2.11. Despliegue para 12 clientes Wi-Fi 6 realizado para la evaluación de desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. 24

2.12. Desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. Evolución de throughput para envíos TCP DL, utilizando 802.11n y ax. Se emplearon los APs Aruba [46] y Cisco Catalyst [43], y clientes con Windows (W) y Ubuntu (U). En 802.11ax, se observa una caída del throughput conforme se incrementan los clientes en la red para ambos APs. Por otro lado, los resultados logrados con el AP Aruba son sensiblemente mejores. 25

2.13. Desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. Evolución de throughput para envíos TCP UL+DL, utilizando 802.11n y ax. Se emplearon los APs Aruba [46] y Cisco Catalyst [43], y clientes con Windows (W) y Ubuntu (U). En 802.11ax, se observa una caída del throughput conforme se incrementan los clientes en la red para ambos APs. Por otro lado, los resultados logrados con el AP Aruba son sensiblemente mejores. 26

2.14. Desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. Evolución de throughput y relación UL/DL para envíos TCP DL y UL+DL, en redes mixtas (con clientes 802.11ax y n en coexistencia). Se empleó el AP Cisco Catalyst [43] y clientes con Ubuntu. En DL, se observa una caída pronunciada del throughput conforme se incrementa la proporción de clientes 802.11ax en la red. En UL+DL la capacidad total de la celda se mantiene más estable y se aprecia que a mayor proporción de clientes 802.11ax, mayor es la proporción de tráfico UL sobre el DL. 27

2.15. Desempeño de IEEE 802.11ax en base a pruebas con equipos comerciales. Evolución de throughput y relación UL/DL para envíos TCP DL y UL+DL, utilizando 802.11n. Se empleó el AP Cisco Catalyst [43] y clientes con Ubuntu. Se observa una caída del throughput en el escenario con 24 STAs, la cual es más pronunciada en UL+DL. En todos los casos predomina el tráfico DL sobre el UL. 28

3.1. Relación entre AI, ML, DL, RL y DRL (adaptada de [53]). 30

3.2. Estructura de una red neuronal artificial simple (adaptada de [59]). 32

3.3. Funciones de activación no lineales popularmente utilizadas (extraída de [60]). 32

3.4. Diagrama de interacción entre agente y entorno de RL (adaptada de [64]). 34

3.5. Diagrama de interacción entre actor, crítico y entorno en el algoritmo Vanilla Actor-Critic (adaptada de [63]). 37

4.1. Topología utilizada y proceso de actualización de CW en el método CCOD (adaptada de [17]). 43

4.2. Diagrama de redes *actor* y *crítico* involucradas en el método CCOD. 44

4.3. Caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Probabilidad de colisión por STA, en promedio y de la red cuando se configura CW óptima para distintas cantidades de STAs. En general, se aprecia un reparto parejo de los valores de esta probabilidad entre todas las STAs de la red. 46

4.4. Desempeño del método CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario estático con $n = 25$ STAs. . 47

4.5. Desempeño del método CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario dinámico con $n = (5; 25)$ STAs. 48

4.6. Desempeño del método CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión y throughput en fase de evaluación, utilizando escenarios dinámicos. Se emplearon distintas cantidades de STAs: $n = (5; 10)$ - fila 1, $n = (5; 25)$ - fila 2 y $n = (5; 50)$ - fila 3. Se aprecian comportamientos oscilatorios en los parámetros y el algoritmo no logra seleccionar una CW cercana a la óptima en ningún caso. . 49

5.1. Desempeño del método E-CCOD para el caso UDP saturado. Valores obtenidos mediante modelo de Bianchi. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario dinámico con $n = (5; 25)$ STAs. 55

5.2. Desempeño del método E-CCOD para el caso UDP saturado. Valores obtenidos mediante modelo de Bianchi. Evolución de CW, probabilidad de colisión y throughput en fase de evaluación, utilizando escenarios dinámicos. Se emplearon distintas cantidades de STAs: $n = (5; 10)$ - fila 1, $n = (5; 25)$ - fila 2 y $n = (5; 50)$ - fila 3. Se aprecia que el algoritmo acompaña la evolución de la red y logra seleccionar una CW cercana a la óptima, siempre que se utilicen cantidades de STAs usadas en la fase de entrenamiento. 56

5.3. Desempeño del método E-CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario dinámico con $n = (5; 25)$ STAs. 57

5.4. Desempeño del método E-CCOD para el caso UDP saturado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión y throughput en fase de evaluación, utilizando escenarios dinámicos. Se emplearon distintas cantidades de STAs: $n = (5; 10)$ - fila 1, $n = (5; 25)$ - fila 2 y $n = (5; 50)$ - fila 3. Se aprecia que el algoritmo acompaña la evolución de la red y logra seleccionar una CW cercana a la óptima, incluso cuando se utilizan cantidades de STAs no usadas en la fase de entrenamiento (aunque en este caso lo hace de manera más ruidosa). 58
5.5. Caso UDP saturado. Valores obtenidos mediante modelo de Bianchi y simulaciones en ns-3. Comparación entre expresiones de probabilidad de colisión, utilizando CW óptima para distintas cantidades de STAs. Se aprecia que los resultados obtenidos con la expresión original y la aproximación de primer orden son muy similares entre sí. 59
5.6. Desempeño del método E-CCOD para el caso con variaciones en el tráfico cursado. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario dinámico donde en la primera mitad de cada episodio se cursó tráfico TCP DL y en la segunda mitad tráfico UDP UL con $n = 25$ STAs. 62
5.7. Formato de trama agregada utilizada para emular envíos MU en ns-3. Cada bloque de datos representa la transmisión de un cliente 802.11ax diferente; mientras que H y T corresponden al encabezado y cola de la trama, respectivamente. 63
5.8. Desempeño del método E-CCOD para el caso de coexistencia entre clientes 802.11ax y *legacy*. Valores obtenidos mediante simulaciones en ns-3. Evolución de CW_{legacy}, CW_{ax}, probabilidad de colisión, throughput y recompensa acumulada en fase de entrenamiento, utilizando un escenario estático con $n = 25$ STAs. 66

Esta es la última página.
Compilado el martes 12 julio, 2022.
http://iie.fing.edu.uy/