UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

Magíster en Ciencias de la Ingeniería c/m en Ingeniería Eléctrica

Profesor Patrocinante:

Phd. Gabriel Saavedra Mondaca.

Comisión:

Dr. Nicolás Jara

Phd. Daniel Sbárbaro.

Dr. Sebastián Godoy

Aplicación de Reinforcement Learning para los problemas de Survivable-Routing, Modulation Level and Spectrum Assigment

Concepción, Agosto de 2022

José Ignacio Núñez Kasaneva

UNIVERSIDAD DE CONCEPCIÓN
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica

Profesor Patrocinante:
Phd. Gabriel Saavedra Mondaca

Aplicación de Reinforcement Learning para los problemas de Survivable-Routing, Modulation Level and Spectrum Assigment

José Ignacio Núñez Kasaneva

Tesis de postgrado
"Magíster en Ciencias de la Ingeniería c/m en Ingeniería Eléctrica"

Concepción, Agosto 2022

Resumen

En el presente documento se propone una solución para el problema de sobrevivencia, enrutamiento, modulación y asignación de recursos (S-RMLSA) dentro de las redes ópticas elásticas (EON) mediante la aplicación de algoritmo de “Deep Reinforcement learning” (DRL).

Las EON corresponden a un esquema de manejo de los recursos de redes ópticas, el cual trabaja con grillas de frecuencia de menor separación comparada con redes tradicionales. Además, permiten el uso flexible del espectro para acomodar conexiones de diferentes tasas, de esta manera flexibilizando y optimizando la asignación de los recursos espectrales de la red.

El problema de RMLSA consiste en la asignación de recursos espectrales, donde el usuario pide requerimientos de bit rate indicando los nodos fuente-destino. Mientras que un algoritmo busca encontrar un formato de modulación y una longitud de onda que sea capaz de transportar todos los datos requeridos en el tiempo que transcurren estas demandas. Las cuales se ven limitadas por los “Frequency slot unit” (FSU) disponibles. Este problema se ha visto solucionados por heurísticas tales como “K Shortest Path” (KSP), “First Fit” (FF).

Por otra parte, la sobrevivencia(S) de la red óptica permite la transmisión de datos aun cuando esta red presente una falla, como la ruptura de un enlace. Lo anterior, indica que el problema a resolver presenta un gran dinamismo por la cantidad de requerimientos que deben ser seleccionados para su envío y los factores externos que pueden afectar la red. Actualmente, estos problemas se solucionan mediante la aplicación de heurísticas, tales como, las protecciones dedicadas 1+1, las cuales necesitan una gran cantidad de recursos para su implementación. Debido al crecimiento de demanda de datos, las redes deben ser capaces de soportar el aumento de tráfico, por lo que la incorporación de nuevas tecnologías y herramientas como la inteligencia artificial proponen una solución al problema de S-RMLSA.

Los algoritmos de “Machine Learning” (ML) son dinámicos y útiles en sistemas de gran complejidad. Estos han sido recientemente implementados dentro de las redes ópticas, modificando el panorama actual, el cual consiste en la aplicación de heurísticas y controladores.

En este trabajo un agente de DRL fue entrenado bajo 3 escenarios de fallas y sin fallas, evaluando su desempeño y comparándolo con diferentes heurísticas de protección y restauración. Se observo una reducción en la probabilidad de bloqueo en 14.99, 26.23 y 53,99% comparado con la heurística de protección y en donde los agentes entrenados con fallas fueron los que presentaron mejor desempeño. Por otra parte, cuando se evalúan los agentes versus la heurística de restauración se obtiene que la heurística supera al agente sin fallas en un promedio del 8 % con respecto a la probabilidad de bloqueo, mientras que en los casos de los agentes con fallas se presenta una mejora promedio del 25.32, 57.62% respectivamente, obteniendo una mejora considerable con respecto a la probabilidad de bloqueo de la red e indicando que entrenando a los agentes en entornos con fallas, estos serán capaces de posicionar los recursos solicitados aun cuando se presenten falla de enlace en la red.

Agradecimientos

Para empezar, quiero agradecer a mi profesor guía de magíster, PhD. Gabriel Saavedra Mondaca el cual me dio la confianza y libertad para poder realizar el trabajo sobre redes ópticas y Deep reinforcement learning, el cual significo un gran reto intelectual, pero al mismo tiempo gratificante por el conocimiento adquirido.

También agradecer al grupo de investigaciones de redes ópticas (IRO) que siempre estuvieron presente y respondieron a todas las dudas que tenía con una excelente voluntad de cooperar. Entregando un conocimiento con el cual siempre contare por el resto de mi vida.

Una mención al grupo de optoelectrónica, con el cual ha sido grato trabajar después de haber estado cerca de dos años de trabajando de manera remota. Generando un entorno agradable para poder aprender y al mismo tiempo reírse.

Agradecimientos al proyecto ANID FONDECYT Iniciación 11190710 sobre “NONLINEARITY COMPENSATION FOR MULTI-BAND TRANSMISSION IN OPTICAL FIBRE COMMUNICATION SYSTEMS”

Finalmente, quiero agradecer a mi familia y amigos que sin importar la distancia siempre han estado presente entregando todo el apoyo y cariño, para que pueda seguir desarrollándome como persona y profesional. Por último, dedico un especial agradecimiento a mi polola Catalina, la cual siempre me ha dado apoyo y amor, representando un pilar estando lejos de casa ¡Muchas gracias por todo¡.

Listado de Figuras

Listado de Tablas

Índice

1. Introducción

El rápido crecimiento de datos en la sociedad moderna ha implicado un constante aumento de tráfico en las redes ópticas, este crecimiento se da por la alta tasa de datos requeridos por los usuarios y la cantidad de ancho de banda que se debe utilizar para poder satisfacer la demanda. Esta expansión de aumento de datos se vio acelerado por la pandemia llegando hasta un 40 a 60%[1] comparado con un año normal. De lo anterior, las proyecciones de consumo de datos que se realizan a las redes actuales quedan desfasadas dado los cambios que se han producido en los últimos 2 años. Esto se ve acrecentado, con la incorporación de nuevas tecnologías tales como 5G, Smart cities. Por lo tanto, el continuo crecimiento de la tasa de consumo provocará una saturación en las redes de fibra óptica, generando indisponibilidad de funciones como aplicaciones que pueden resultar ser esenciales. Este crecimiento de la infraestructura como en sus proyecciones se puede visualizar en Figura N°1, que nos indica la tasa de crecimiento de infraestructura de redes ópticas en distintas partes del mundo en el último año.

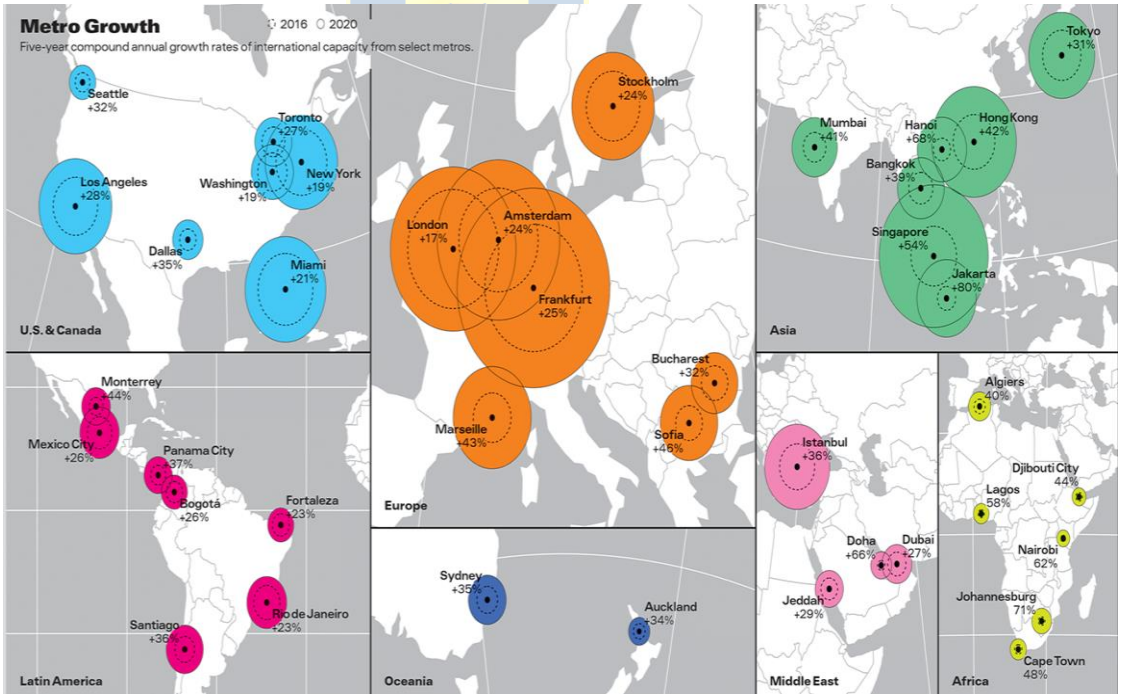

Figura N°1: Figura de proyecciones de aumento de las infraestructuras en las redes ópticas del mundo[2]**.**

Actualmente, el modelo que se utiliza para regular los recursos utilizados por las redes ópticas corresponde a “Wavelength division Multiplexing” (WDM), la cual trabaja con FSU predefinidas de un ancho de banda fijo de 50GHz para multiplexar señales en frecuencia. El esquema de WDM, corresponde a uno de los primeros esquemas utilizados para administrar y asignar los recursos espectrales[3], este esquema de trabajo no permite una optimización global de los recursos ópticos utilizados de la red, dejando ancho de banda sin utilizar entre los canales, generando una fragmentación del espectro, la cual consiste en anchos de bandas espectrales que no están siendo utilizados dado que no han ocupado posiciones contiguas, lo cual provocará que no se optimicen estos recursos generando problemas a mediano a largo plazo, tales como el “Capacity Crunch”[4].

De lo anterior, se proponen las “Elastical Optical Networks” (EON)[5]. Las EON trabajan con separación espectral de menor tamaño (12.5 GHz o 6.25 GHz) y flexibles, lo que permite agrupar las solicitudes en caso de que se requiera un mayor espectro, utilizando de manera más eficiente el espectro para la asignación de recursos, disminuyendo la fragmentación[6]. Esta diferencia de los esquemas WDM y EON se visualiza en la Figura N°2.

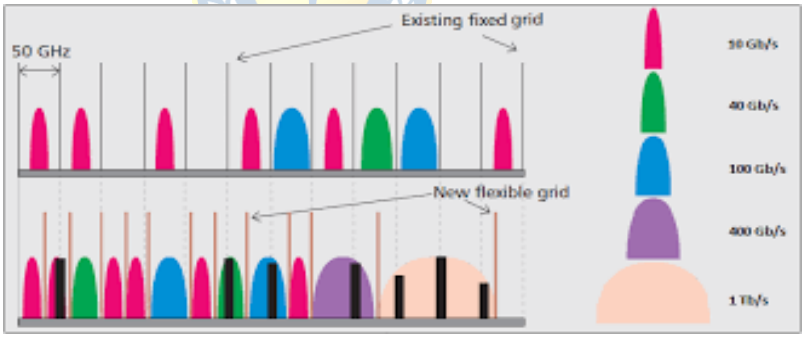

Figura N°2: Asignación de espectro en grillas de 50GHz a grillas de 12.5GHz, 6.25GHz [5].

El “Capacity Crunch” consiste en un problema en donde los recursos de la red, tales como la cantidad de espectro disponible se ven insuficiente para soportar la cantidad de demanda requerida, reduciendo la calidad de servicio entregada[7]. De lo anterior, en caso de que no se tomen medidas pertinentes significará una red saturada en horarios peak, en donde las personas se adaptan a los patrones de disponibilidad o sencillamente un colapso en la red dada la gran cantidad de solicitudes de espectro requeridas. De cualquier forma, esto significa

un constante deterioro de la calidad del servicio llegando al punto en el cual los servicios más críticos se ven afectados. Otro factor que puede provocar una disminución en la calidad de servicio de la red corresponde a las fallas dentro de las redes ópticas. Dado la cantidad de factores externos que pueden generar falla, pasan a ser impredecibles, por lo cual siempre existirán fallas y se deberá tener en cuenta en el diseño de la red. En las redes ópticas, se generan cuatro posibles fallas, la primera corresponde a la falla de enlace (link), correspondiente a un enlace cortado dado por un motivo externo o saturación. La segunda causa corresponde a las fallas de camino (path), la cual consiste en una falla de tramos y múltiples enlaces, las fallas de nodo, en el cual todos los enlaces de ese nodo se ven afectado por las fallas, y por último son las fallas de FSU (o canales ópticos), en donde un transceiver falla y no se puede utilizar el canal óptico requerido.

Un punto por destacar en los esquemas WDM y EON, es que, en caso de falla de un enlace o varios, deben ser capaces de mantener la transmisión entre los usuarios. Para impedir que estos servicios se vean interrumpido, se necesita tener una infraestructura que pueda reestablecer rápidamente la comunicación entre los nodos afectados, lo anterior corresponde a la “tolerancia de falla” [8]. También, la ocurrencia de fallas con las que las redes ópticas se ven afectadas, en el reporte [9] indica que existe un corte de fibra óptica cada 5 dias en la red NSFNET[10], por otra parte, se indica que el tiempo que la red NSFNET está sin poder entregar servicio por año corresponde a 24 hrs [11]. Lo cual para transmisión de grandes volúmenes de datos es una perdida masiva y la cual se debe tener en cuenta al momento del diseño de las redes ópticas.

Los esquemas de las EON y WDM tienen la capacidad de trabajar en sistemas dinámicos como estáticos. Una de las principales diferencias que existe entre los sistemas dinámicos y estáticos, es que para el caso dinámico los recursos son otorgados solo cuando son requeridos por los usuarios, mientras que en el caso estático los recursos se otorgan de manera permanente en cada conexión. En ambos esquemas, la cantidad de información transmitida es alta (llegando al orden de los Tb/s). Para el caso de las EON, debido a que presentan grillas flexibles de menor tamaño que en el caso de las WDM[12], la información que pueden transmitir aumenta ya que disminuyen los efectos de la fragmentación del espectro óptico[13], por lo que en caso de alguna interrupción ya sea por saturación de uno de los enlaces, como de

alguna ruptura del enlace generaría una perdida masiva de datos. Dado la masividad de redes ópticas que se están ocupando y como el crecimiento continuo de estas redes sigue, pone un antes y un después en la planificación de las redes, ya que requieren de un sistema de protección para que la información pueda llegar a los nodos requeridos en el caso que existan fallas[14].

La sobrevivencia de las redes ópticas corresponde a la función de poder reestablecer la transmisión de la información dentro de la red aun cuando uno de sus tramos se vea bloqueado o cortado por factores externos. Las técnicas de protecciones de las redes ópticas para la sobrevivencia pueden ser caracterizadas en "Link-Based" y las "Path-Based" [15]. Para el caso del "Link-based" recupera el tráfico de la red rodeando el "link" donde se genera la falla, mientras que la técnica "path-based" recupera la falla en los dos nodos finales de una ruta. Las técnicas de protección de la red también se caracterizan en función de las topologías de red, en donde se clasifican en protecciones basadas en "Ring-based protecction" y "Mesh-based protecction". Para el caso del "Ring-based protecction" se dividen en las "Unidirectional Path Switched Ring" (UPSR) y "Bidirectional Line Switched Ring" (BLSR). Por otra parte, el "Mesh-based protecction" ha empezado a tener más atención en los últimos años debido a su eficiente intercambio de capacidad, aun cuando presenta una alta complejidad topológica. Lo anteriormente mencionado se visualiza en la Figura N°3 indicando los principales métodos utilizados para la protección y sobrevivencia de las redes ópticas.

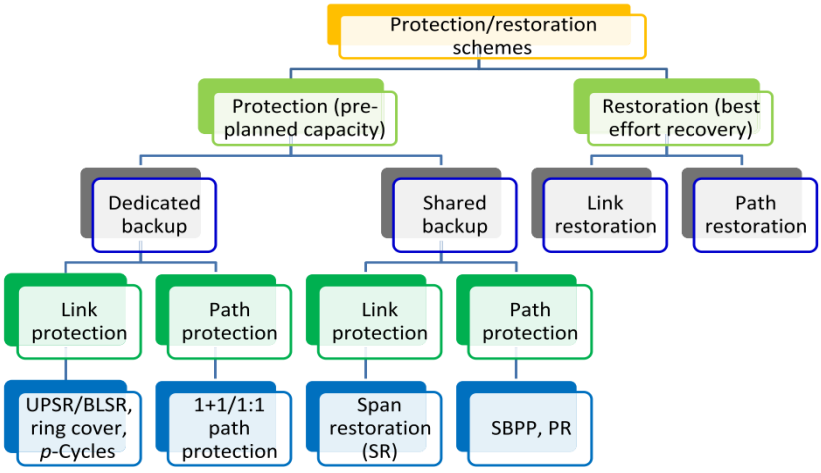

Figura N°3: Esquemas de protección de redes ópticas[15].

De la Figura N°3, también se observa que existe el proceso de recuperación una vez ocurrida la falla, este proceso tiene por finalidad reestablecer los servicios perdidos dado por la contingencia que presento la red óptica. Existen dos maneras de recuperar los servicios perdidos y ambos comparten un buffer en donde guardan la información transmitida y los nodos de fuentes y destino. La primera técnica corresponde a la restauración de enlace en donde se bordea el enlace con falla mediante los FSU disponibles por los enlaces aledaños, por otra parte, existe la técnica de restauración de tramos, en donde mediante rutas aledañas se pueda bordear la falla, estas rutas aledañas no presentan enlaces compartidos por lo que pasan a ser rutas disjuntas. Los anteriores métodos mencionados nos entregan la posibilidad de resolver el problema de sobrevivencia(S) dentro de las redes ópticas

Las redes ópticas están conformadas por distintos enlaces en los cuales cada uno tiene características en específicos, estas caracteristicas corresponde al formato de modulación aplicado[16][17], la longitud del enlace, la continuidad del canal óptico seleccionado en las rutas de trabajo, la cantidad de FSU ocupados y libres para asignar en el espectro óptico, la demanda requerida por los usuarios y la ruta seleccionada para él envió del mensaje. Lo anterior expuesto, corresponde a un problema de enrutamiento y selección recursos el cual lleva el nombre de “Routing, modulation formats-level and spectrum assigment” (RMLSA)[18][19] y en donde para establecer las conexiones en una EON cada usuario se le debe asignar RMLSA. La solución de este problema, puede ser complicado, debido a la gran cantidad de variables que afectan el enrutamiento de la conexión, es por lo anterior que existen una gama de soluciones dentro de las cuales está la utilización de “Software defined networking” (SDN)[20] [21], la cual define mediante un software la distribución de estos recursos y enrutamiento que debería tener calculando los FSU necesarios y el formato de modulación dentro de la red óptica, comportándose como un gran controlador de la EON. Si al problema RMLSA, se plantea un enfoque se sobrevivencia de la red, este problema pasa a ser S-RMLSA.

Las redes ópticas se planifican para poder operar bajo una cantidad de demanda de la red, y poder establecer la mayor cantidad de conexiones posibles sin que se presenten saturaciones[18][22]. Aun así, está planificación no fue pensada para el continuo crecimiento que están se están presentando las redes, por lo cual nuevamente va a existir el problema de “Capacity Crunch” y un empeoramiento de la calidad de servicios.

Dado el masivo uso de las redes ópticas y la cantidad de información que transportan, el problema de S-RMLSA a las redes debe ser tomado en cuenta y ser uno de los principales problemas a resolver, ya que las heurísticas actuales se ven sobrepasadas por la cantidad de requerimientos y aunque exista un sobredimensionamiento para trabarlas, ante tasas de crecimiento de 40% anual, significa una constante carga de tráfico que impedirá el correcto funcionamiento de la red [23].

Debido a los avances en las tecnologías y las capacidades computacionales se están probando nuevas técnicas para resolver el problema de S-RMLSA. Estos nuevos métodos utilizan las ventajas del procesamiento de grandes volúmenes de datos y su versatilidad para el control de escenarios dinámicos. Por lo anterior, se está empezando a utilizar los métodos de "Machine Learning" (ML) para la administración de estos recursos de la red[24]. Por otra parte, debido a que se busca el funcionamiento óptimo de la red se aplica una de las ramificaciones del ML denominada de "Reinforcement Learning" (RL), la cual trabaja bajo entornos y agentes que interactúan entre si permitiendo el control de los recursos de la red mediante la toma de acciones del agente. En donde un agente utiliza grillas flexibles y posiciona a los paquetes de información enviados, mientras que otro agente verifica que la señal enviada pueda llegar a su destino final [23].

Para poder trabajar con estos agentes, se deben generar entornos que tengan la capacidad de entregarles los escenarios más reales para que estos agentes puedan entrenar. Es por lo anterior, que los autores de [25] crearon el entorno "DeepRMSA" el cual crea agentes de SDN que permiten la administración de recursos de la red mediante la aplicación de las heurísticas KSP-FF como base en sus agentes SDN, esto con el fin de solucionar el problema de RMLSA.

En el presente trabajo se pretende expandir las capacidades del entorno "DeepRMSA" para esto se crean nuevas funciones capaces de recrear escenarios con fallas generando un nuevo entrono denominado "DeepSRMLSA" que será especificados más adelante en el documento. En este nuevo entorno, el agente brindará un enfoque de sobrevivencia a la EON, de tal manera de poder solucionar el problema de S-RMLSA y que actuará bajo restauración

de falla de enlace, es decir, de una manera retroactiva. Para poder llevar a cabo este planteamiento, al agente se le entregará una gran cantidad de caminos disponibles y en el cual podrá elegir donde posicionar el espectro, dado que su espacio de acción corresponderá al de la heurística KSP-FF. Además, en este articulo será la primera vez que un agente de DRL es entrenado con fallas de manera secuencial según la cantidad de enlaces que presente la red, ya que en el proceso de entrenamiento se generará una falla por cada enlace de la red NSFNET de tal manera que el agente aprenda a tener una política de acción ante una falla y de esta manera salir de los típicos modelos en el cual al agente se le entrena con una espacio de acción del agente correspondiente a una heurística en particular. Esta aplicación después será evaluada con el algoritmo KSP-FF correspondiente al espacio de acción con el que se entrena los agentes, el cual nos entrega la restauración que se podrá obtener tanto por la heurística como por el agente y con una heurística con enfoque de sobrevivencia correspondiente al algoritmo 1+1 con esquema de protección dedicada (que es presentando en la sección 6.2) . Estas heurísticas y agente serán comparados bajo las mismas métricas para visualizar el rendimiento del agente ante distintos escenarios.

2. Hipótesis

Los algoritmos de “Reinforcement Learning”, permiten la optimización del uso de recurso de la red y protección de las “Elastical Optical Network” resolviendo el problema de “Survivable- Routing, modulation level and spectrum assigment”.

3. Objetivos

3.1. Objetivo General

Implementar algoritmo de Reinforcement Learning para la solución del problema de RMLSA y survivability en EONs.

3.2. Objetivos Específicos

1. Habilitar el OpticalGYm para la implementación y estudio de las EON.
2. Simulación de fallas en enlaces de redes ópticas en el ambiente GYM.
3. Incorporación de algoritmo en el agente para resolver el problema de RMLSA y survivability (S-RMSLA).
4. Implementar la evaluación y la heurística.
5. Análisis de los resultados.

3.3. Limitaciones del trabajo.

- Las simulaciones se realizarán en el lenguaje de programación Python dado la gran variedad de librerías y OpemGYM que se presentan.
- Solo se trabajará con un agente de DRL, correspondiente al agente Actor Critic (A2C).
- Los entrenamientos y evaluaciones se realizarán en Google Colab.
- Los resultados de las evaluaciones serán comparados con el comportamiento del agente en su entrenamiento, para la visualización del aprendizaje obtenido.
- Se trabajará bajo cargas de tráfico uniformes tanto para el trabajo de estado de la red sin fallas como con falla de enlace.
- Las fallas se simularán como saturaciones de los enlaces de la red.

4. Estado del Arte.

En los reviews, “Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies,” [13] y “Survivable elastic optical networks: survey and perspective (invited),” [15]. Hablan tanto de los principales problemas que presentan en las redes ópticas con los esquemas WDM y EON, tales como el problema RMLSA y S-RMLSA dadas por las limitantes que se expresaron en la sección 1. En donde se indican distintas soluciones que se aplican tales como las heurísticas KSP-FF, Random Fit o Best fit y las desventajas que presentan estas heurísticas, que en el caso del algoritmo KSP, muchos de los caminos seleccionados corresponden a enlaces con la menor longitud posible, por lo que estos enlaces más cortos se empiezan a repetir en la selección de rutas, dejando vulnerable la red en caso de fallas en estos enlaces.

El área de sobrevivencia de las redes ópticas ha sido un área abordada por distintos investigadores, en los cuales plantean distintas soluciones, una de ellas corresponde al trabajo “A novel shared-path protection algorithm with correlated risk against multiple failures in flexible bandwidth optical networks,” [26], el cual utiliza protecciones compartidas (SPP) ante múltiples fallas de la red, en donde las SPP, los recursos espectrales son asignados en una ruta primaria, y al mismo tiempo se calcula una segunda ruta en donde los recursos espectrales pueden ser enviados en caso de fallas, pero con la condición que solo será activado una vez que se detecte una falla, es por lo anterior, que estas segundas rutas son compartidas con distintos usuarios. El SPP puede ser ejecutada de dos maneras, en donde la primera correspondiente a la off-line SPP, la cual corresponde a rutas calculadas antes de la operación de la red, mientras que para el caso on-line las rutas de protección son calculadas una vez que la falla ocurre, por lo que se debe calcular cada vez que ocurre una falla.

Siguiendo en el área de Sobrevivencia (S-RMLSA) se presenta el trabajo llamado “Protection in elastic optical networks using Failure-Independent Path Protecting p-cycles,”[27]. El cual utiliza dos rutas, la primera corresponderá a la asignación de los recursos ópticos dada por la demanda del usuario y la segunda corresponderá a una ruta de protección de recursos dedicados, es decir, que estos recursos solo se reservan para que, en caso de falla, poder enviar los requisitos solicitados por el usuario por este segundo camino manteniendo una

sola longitud de onda. Otra característica que presenta esta heurística es que las rutas de protección calculadas toman formas de anillos de tal manera de bordear la zona donde ocurre la falla y de esta manera redistribuir los recursos ópticos. La desventaja que presenta es que al momento de poder aplicar esta heurística dependerá del tamaño de la red ya que puede que en redes grandes el reach entregado por el formato de modulación no sea lo suficiente y se genere un bloqueo, por lo que requerirá una reconversión de longitud de onda, generando delay en la transmisión de la información.

El trabajo mezcla presenta conocimientos de ciencia de datos enfocado en Reinforcement learning, para lo cual se utilizan los siguientes papers "Multi-Agent Reinforcement Learning for Problems with Combined Individual and Team Reward", "Deep Q-Network Based Multi-agent Reinforcement Learning with Binary Action Agents" y "Dual-agent deep reinforcement learning for deformable face tracking" [28]–[30], respectivamente, los cuales indican como implementan sus agentes a distintos entornos y aplican la técnica de Transferencia de conocimiento (TL) para que el aprendizaje de los agentes en los ambientes utilizados sea maximizado. Para esto, explican diferentes métodos de entrenamiento para aplicar el TL y que el agente pueda presentar buenos resultados en sus evaluaciones, los cuales son utilizados en el presente trabajo y será explicado en la sección 6.1.

La incorporación de ciencia de datos con las redes ópticas ha ido en aumento estos últimos años, estos avances se pueden ver en los review "Machine learning for intelligent optical networks: A comprehensive survey" [31] y "Overview on routing and resource allocation based machine learning in optical networks" [32]. Los avances presentados convergen a puntos en común en donde la aplicación de técnicas de ML se ven enfocadas en la estimación de la calidad de la señal, prevención de distintos escenarios, mediante técnicas tales como Cluster, SVM que nos permiten agrupar o clasificar según los parámetros que necesitemos distintas situaciones que proponemos. Una de las principales conclusiones que se obtiene por los anteriores dos review es que desde 2019, han salidos investigaciones sobre el desarrollo y control de las EON mediante la asignación de recursos, utilizando Deep Reinforcement learning (DRL) aprovechando las ventajas de las redes neuronales, aplicado en el aprendizaje automático para el control de las EON para solucionar el problema de RMLSA o RSA[25], el cual crea un entorno denominado "DeepRMSA" y mediante el espacio de acción

basado en KSP-FF asigne los recursos ópticos y demostrado tener una ventaja sobre las heurística con la que se basa su espacio de acción, aun así este trabajo no soluciona el problema de la sobrevivencia de las redes pero es una primera aproximación importante para el desarrollo de futuras investigaciones.

Uno de los principales trabajos que han sido publicados y que solucionan el problema S-RMLSA, corresponde a [23], el cual aplica dos agentes de DRL, mediante una aplicación de redes convolucionales(CNN), las cuales extraen mediante la matriz de característica de la red y en el primer agente de DRL asigna una ruta junto con el formato de modulación para poder enviar la información, mientras que el segundo agente, mediante un espacio de asignación dado por el esquema de de "$\rho - cycle$" va encontrando las rutas de protecciones dedicadas que podrán enviar la información, este esquema de protección presenta excelente resultados comparado con heurísticas de protección, pero la desventaja es que utiliza un alto número de espectro en la ruta de protección dedicada, las cuales podrían ser ocupadas por otros usuarios, por lo que se necesita al menos dos veces o más el espectro necesario para poder enviar la información. Por otra parte, los trabajos, [33], [34] nos entregan un esquema de DRL para la solución del problema S-RMLSA, pero con un enfoque de restauración dentro de las redes WDM, por lo que las demandas de los usuarios son estáticas, en este trabajo, mediante la implementación de CNN, crean una distribución con los caminos disponibles de la red con la cual, al momento de presentar alguna falla, sea capaz de restaurar la mayor cantidad de servicios interrumpidos, las desventajas de estos trabajos, es que dado que el esquema es estático, ante demandas de usuarios de manera aleatoria no será capaz de procesar y podría generar saturaciones, esto debido a que el agente toma una imagen de cómo está distribuida la red y sabe la cantidad de servicios que necesitara entregar.

Dentro de las aplicaciones de DRL dentro de las redes ópticas, el trabajo de los autores [35] consiste en aumentar la capacidad de los canales mediante la incorporación de multibandas e implementando algoritmos de selección de distintas bandas. También, en contra parte, existen trabajos de los autores de [36], los cuales asignan los recursos en fibras multinúcleos expandiendo las capacidades de transmisión y disminuyendo el capacity brunch.

Debido a que se quiere expandir las ventajas de los algoritmos de ML sobre las EON,

se utiliza el framework de código abierto OpenGym presentado en [37], lo cual permite la expansión de las funciones que se le puede agregar al agente DRL en el control y optimización de las EON. Esto último, es fundamental para la realización y expansión en el problema de "S-RMLSA", ya que los distintos algoritmos de RL (A2C, PPO2, DQN) pueden manejar grandes volúmenes de información y ser capaz de optimizar los recursos protegiendo las redes ante los distintos escenarios que se le puede plantear.

Lo anterior mezcla los conceptos de protección de EON [15] y conceptos de ciencia de datos[38], generando una poderosa herramienta que se puede utilizar para abordar el problema. Como se mencionó anteriormente solo existe un trabajo que trata de solucionar el problema de S-RMLSA dentro de las EON, por lo tanto, la realización de este proyecto implica un desarrollo y con un impacto dentro de las investigaciones relacionadas al área.

La aplicación de ML a los ambientes de redes ópticas generará nuevas maneras para la distribución y manejo de los recursos de las EON[24], agregando maniobrabilidad [39] y una nueva manera de optimizar los recursos de la red como de sus futuras aplicaciones [40],[41].

5. Metodología

5.1. Problema de "Survivable-Routing, Modulation, Spectrum,Assignment" (S-RMLSA).

Para complementar el estado del arte, el problema de S-RMLSA corresponde a una de las dificultades más importante a resolver en las EON, que tiene como objetivo satisfacer de manera efectiva las diversas demandas de los usuarios con la garantía de sobrevivencia de la red, mientras que el problema de RMLSA se utiliza para encontrar las rutas de trabajo más adecuada para un par de nodos de fuente-destino, y la correcta utilización del espectro óptico para satisfacer las demandas de los usuarios [23].

Las EON transportan grandes volúmenes de información (en el orden de los Tb/s), es por lo anterior que la sobrevivencia de las EON es fundamental, ya que en el caso de una interrupción se producen masivas perdidas de datos. Por lo que, en la actualidad existen distintas maneras de abordar este problema [26], [42]–[44] para que la transmisión pueda seguir aun con las fallas.

Por otra parte, el problema de RMLSA es un problema que ha sido abordado en el caso de sobrevivencia, y al igual que en el caso anterior existen varias maneras de solucionarlo[18][45]. Pero uno de los principales métodos corresponde a SDN [20]. El cual mediante un software se define como va a actuar la red ante una falla, según los recursos que esta va disponiendo y visualizando la red de manera global, generando un dinamismo para los requerimientos de la red, lo anterior se puede visualizar en la Figura N° 4.

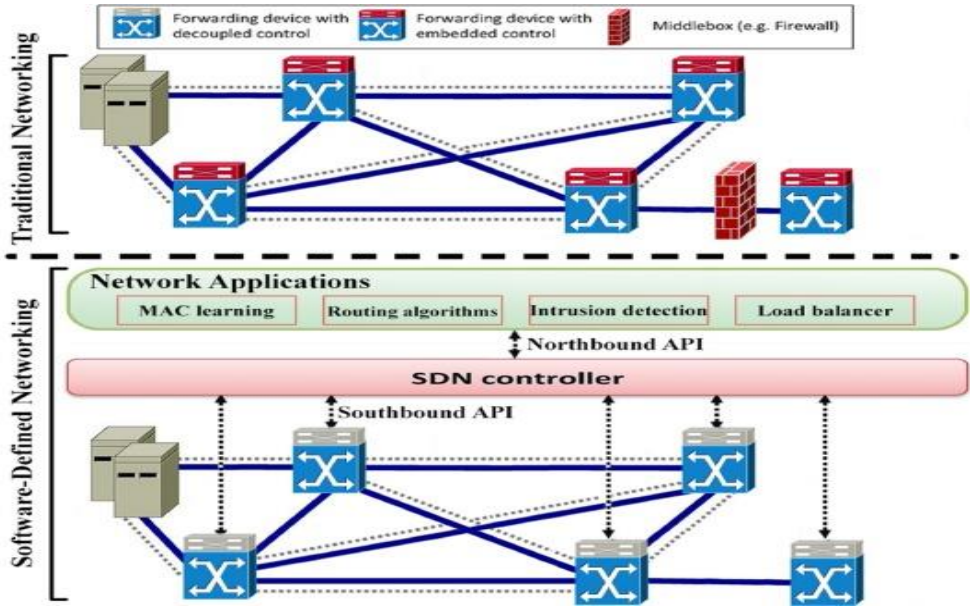

Figura N° 4: Esquema de trabajo del Controlador de SDN[46].

La correcta implementación para solucionar el problema de S-RMLSA permitirá una optimización de las EON, resolviendo los requerimientos de los usuarios y reaccionando en tiempos acotados en casos de fallas, como también disminuyendo los costos de capital humano utilizado para el manejo de las redes. Esta correcta implementación permitirá mejorar el funcionamiento de la EON, permitiendo una disminución en las saturaciones que existen en las redes ópticas. Pero para poder llevar a cabo este objetivo, se necesita tener una herramienta poderosa que sea capaz de incorporar grandes volúmenes de información, que además sea dinámica como flexibles en cuanto al procesamiento de la información y que los tiempos de reacción de esta herramienta sean acotados. Por lo mismo, es que este problema ha sido abordado mediante las técnicas de ML[23].

Estas técnicas de ML que actúan sobre el problema de distribución de recursos dentro de una red son basadas inicialmente para la distribución y protección de redes de Datacenters, en el cual aplican algoritmos de RL para entregar una protección en caso de fallas y una correcta operación para la distribución de datos entre los distintos nodos [47].

En las siguientes subsecciones 5.1.1, 5.1.2 y 5.1.3 se expondrá el modelamiento la formulación, el principio de operación del entorno del agente y las métricas que se utilizarán para poder evaluar el desempeño del agente.

5.1.1. Formulación de S-RMLSA

Se toma la función $G(V, E, F)$ la cual denota la topología de la EON, en donde V y E representan el set de nodos y links respectivamente, además $F = \{F_{e,f}|_{e,f}\}$ contiene los estados de las FSU, en donde $f \in [1, f_o]$ correspondiente a la cantidad de FSU de cada enlace, y $e \in E$ corresponde a los nodos con los que se está trabajando. Los requerimientos de los caminos recorridos desde el nodo o al nodo d $(o, d \in V)$, se escriben como $R_t(o, d, b, \tau)$, donde b corresponde al bit rate requeridos en Gb/s y τ corresponde al ancho de banda solicitado. Por otra parte, Para dar los requerimientos necesarios a R_t, se necesita enrutar un camino "end to end" $(P_{o,d})$, en el cual se pueda determinar un formato de modulación m adecuado para asegura la calidad de transmisión QoT y posicionar los números de espectros FSU contiguos en cada enlace de acuerdo con $P_{o,d}$, en donde P correspondera a la ruta seleccionada. En el trabajo se asume que la EON no presenta una reconversión de espectro, por lo tanto, se presenta una restricción en cuanto a la continuidad de espectro en $P_{o,d}$. Además, el número de FSU necesario para establecer una conexión se calcula mediante la siguiente expresión.

$$n = \left[\frac{b}{m * C_{grid}^{BPSK}}\right] \quad \textbf{(1)}$$

En donde C_{grid}^{BPSK}, corresponde a la tasa de transmisión que un FSU puede soportar, $m \in [1,2,3,4]$ corresponden a los formatos de modulación BPSK, QPSK, 8-QAM y 16-QAM respectivamente y b corresponde al bit rate requerido. Para el caso del problema de RMLSA estático presenta requerimientos estáticos por lo tanto $R = \{R_t|_t\}(\tau \to \infty)$ y requiere aprovisionarlos todos los FSU en un lote siguiendo las restricciones de capacidad del enlace. Por lo tanto, en el caso estático del problema de RMLSA el objetivo es minimizar el total de espectro utilizado.

En una EON dinámica, los requerimientos de FSU son usados y liberados en distintos instantes de tiempo, por lo tanto, el objetivo del problema de RMLSA es minimizar a largo plazo la probabilidad de bloqueo (PB) presentada en la red, lo anterior se define como la tasa de conexiones rechazadas sobre conexiones, la función a minimizar se expresa en (2) y la ecuación que se aplica en (3).

$$\min BP \tag{2}$$

$$BP = \sum_{i}^{N} \frac{\left(N_i - \sum_{j}^{t} I_{ij}\right)}{N_i}, t \in [t_o, .., T] \ \wedge \ i \in [1, .., N] \tag{3}$$

En donde N_i corresponde al número total de solicitudes que ocurren en un tiempo t, mientras que I_{ij} corresponde al numero de solicitudes aceptadas por la red. En caso de que se requiera saber el impacto que tiene la cantidad de conexiones interrumpidas por ancho de banda queda, queda expresada como:

$$BP_{BW} = \sum_{i}^{N} \frac{\left(N_i * \tau_i - \sum_{j}^{t} \tau_j * I_{ij}\right)}{N_i * \tau_i}, t \in [t_o, .., T] \ \wedge \ i \in [1, .., N] \tag{4}$$

En el presente trabajo se definirá la falla de un enlace como la saturación de este, la cual queda definida en la siguiente expresión:

Para el caso dinamico sin *fallas, los FSU disponibles quedaran expresados*

$$\{F_{e,f}|_{e,f}\} = 1$$

Mientras que para el link seleccionado donde se generará la falla queda expresado como:

$$\{F_{link,f}|_{link,f}\} = 0$$

Indicando la indisponibilidad del enlace para poder alojar nuevas conexiones.

5.1.2. Principio de operación del entorno "DeepRMLSA"

La Figura N°5 explica el principio de operación del entorno de [25] para la solución del problema de RMSA, el cual toma ventaja del SDN, como un centro automatizado de control para el manejo de las EON, lo cual se explicará en las siguientes etapas.

Etapa 1: Un controlador remoto interactúa con los agentes SDN locales (ubicados en los nodos) los cuales recopilan los estados de la red, las solicitudes de los usuarios y el esquema RMLSA que se está ocupando en la EON en el periodo de tiempo t. Este entorno utiliza un controlador de agentes SDN, con las cuales van configurando las conexiones de los nodos de acuerdo con los comandos recibidos de la red neuronal del DRL, al recibir una solicitud de ruta de luz Rt.

Etapa 2: El controlador SDN recupera de la base de datos de tráfico las representaciones clave del estado de la red, incluidas las rutas de luz en servicio, la utilización de recursos y la abstracción de topología, e invoca el módulo de ingeniería de características para generar datos de estado personalizados para el entorno de "DeepRMSA".

Etapa 3: La red neuronal densa (DNN) del entorno leen los datos de estado y generan una política RMLSA de operación $\pi_t(A, |s_t, \theta)$ para el controlador SDN, donde A es el conjunto de esquemas RMLSA candidatos para Rt (correspondiente a la acciona tomar) y θ representa el conjunto de parámetros de la DNN.

Etapa 4: π_t entrega una distribución probabilística en A. El controlador toma acciones $a_t \in A$ basada en la política π_t y selecciona un correspondiente camino a tomar.

Etapa 5: El sistema recibe el retorno previo del esquema RMSA anterior a la nueva acción.

Etapa 6: Se produce una recompensa inmediata r_t, y en donde la recompensa anterior r_{t-1}, junto el estado de la red s_t y la acción de la red a_t, son guardados en un buffer, generando una recopilación de estado, acciones y recompensas por acciones tomadas.

Etapa 7: Las señales de entrenamiento actualizan la DNN, la cual tiene por objetivo maximizar a largo plazo la recompensa obtenida.

Las etapas anteriormente explicadas presentan la interacción del agente DRL con el entorno “DeepRMSA”. Indicando la capacidad de aprendizaje, dado por la exploración de acciones, las cuales retornan recompensas y posicionan al agente en distintos estados del entorno, generando política de acción dentro del agente que servirá a futuro para su implementación y posterior evaluación.

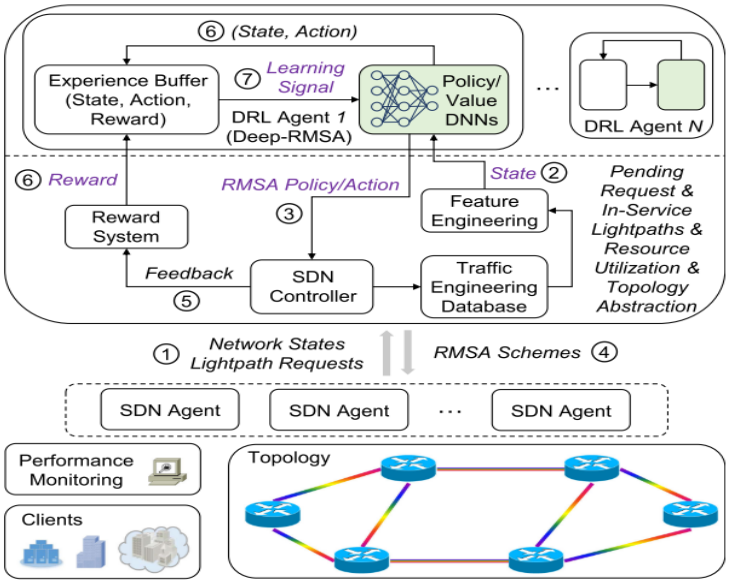

Figura N°5: Principio de Operación del entorno “DeepRMSA” [25].

5.1.3. Modelamiento y entrenamiento del agente.

Se define la representación de estados del agente para la lectura de los resultados, las acciones que el agente puede tomar y las recompensas que obtiene, además de explicar las caracteristicas dinámicas de provisionamiento. Además, se expondrán las métricas con las que se evaluará el comportamiento del agente y como este agente se desempeña en el ambiente.

Las métricas presentadas se obtendrán mediante una media móvil de 50 puntos entre ellas, y corresponden a:

- **“Service blocking rate”:** Corresponde a la probabilidad de bloqueo de la EON dada por la ecuación (3) y tomando su media móvil cada 50 iteraciones, entregando el desempeño general de la red.
- **“Bit rate blocking rate”:** Retorna la probabilidad de bloqueo por ancho de banda de la EON dada por la ecuación (4) con una media móvil de 50 iteraciones.

- **"Service blocking rate per episode":** Retorna la probabilidad de bloqueo de la EON por episodio de entrenamiento entregada por la ecuación (3), entregando los cambios dinámicos que existen dentro de su operación
- **"Bit rate blocking rate per episode":** Retorna la probabilidad de bloqueo por ancho de banda por episodio de la EON dada por la ecuación (4).
- **"Users Counter":** Retorna la cantidad de usuarios solicitando los servicios.
- **"Users accepted":** Retorna los servicios de usuarios aceptados.
- **"Path":** Retorna los caminos que usa el agente para asignar para enviar los datos.

Las anteriores métricas mencionadas nos otorgarán un rendimiento del agente para los entornos tanto de entrenamiento y evaluación.

5.1.3.1. Modelamiento del entorno de la EON

1) **Entorno**: La representación del entorno s_t que el agente va a obtener como entrada corresponderá a una tupla de tamaño $1\,x\,(2|V|+1+(2J+3)k)$, conteniendo la información de R_t y la utilización de FSU dentro de los K caminos seleccionados, esta tupla corresponde a:

$$s_t = \left\{o, d, \tau, \left\{\left\{z_k^{1,j}, z_k^{2,j}\right\}\right\}|_{j\in[1,J]}, z_k^3, z_k^4, z_k^5\right\}|_{k\in[1,K]}\} \quad \textbf{(5)}$$

La primera etapa de la tupla presenta 2|V|+1 elementos, en donde se convierten los nodos fuentes (o) y destino(d), en un formato "one hot format". Además, τ indica el bit rate solicitado, donde |V| indica el número de nodos en V. Por otra parte, $z_k^{1,j}$ calcula la cantidad de FSU disponibles en donde puede ser enviada la solicitud de conexión, mientras que, $z_k^{2,j}$ indica el índice donde serán posicionados los FSU dentro del espectro disponible, "J" indica el numero de FSU asignados (este último estará de acuerdo con el formato de modulación aplicado). Además, $z_k^{3,j}$ entrega el número de FSU requeridos, por otra parte, $z_k^{4,j}$ corresponde al promedio de FSU disponible, y para

finalizar, $z_k^{5,j}$ nos indica el número total de FSU disponibles. Lo anterior expuesto será explicado con un ejemplo ilustrativo.

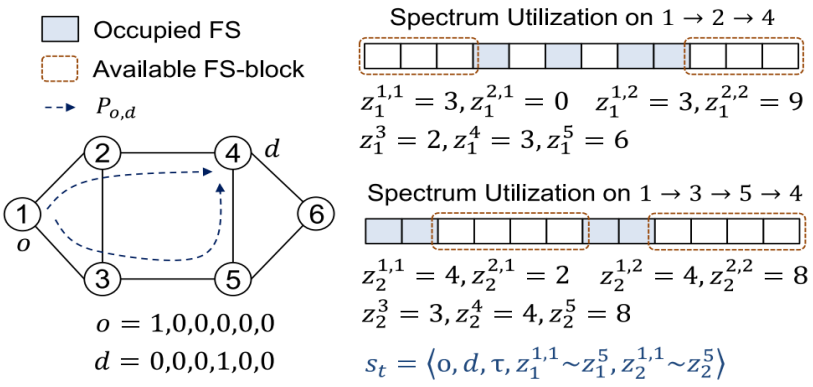

Figura N°6: Ejemplo ilustrado en la Asignación de espectro y selección de rutas dentro del entorno DeepRMSA [25]**.**

La Figura N°6 propone dos ejemplos de asignación de rutas, en donde se desea asignar una conexión desde el nodo 1 hasta el nodo 4, para lo cual se proponen dos caminos y se necesitan dos FSU, es decir, $k = 2$ y $J = 2$ respectivamente, en donde el primer camino recorre por los nodos 1→2 → 4 mientras que el segundo camino recorre los nodos 1→3→5→4.

Para el primer caso, se visualiza los nodos 1 y 4 que están representando en un formato "one hot format", además, los parámetros de la tupla $z_1^{1,1} = 3\ y\ z_1^{2,1} = 0$ nos indican la cantidad de FSU disponibles donde puede ser enviada la conexión y la posición de estos FSU en el espectro de la ruta respectivamente, pasa lo mismo para el caso de la segunda ruta $z_1^{1,2} = 3\ yz_1^{2,2} = 3$. Por otra parte, como se mencionó anteriormente $z_1^3 = 2$ corresponde a la cantidad de FSU necesaria para enviar la información, indicando que se necesitan 2 FSU para la trasmisión de datos por ese camino. Además, $z_1^4 = 3$ entrega la media de FSU disponibles en donde se puede enviar la información y para finalizar $z_1^5 = 6$ corresponde al total de espectro disponible para enviar las conexiones. Para el segundo caso (camino 1→3→5→4) es el mismo procedimiento, pero con la diferencia que el formato de modulación aplicado es distinto, por lo que, $z_2^3 = 3$ a diferencia del primer camino en donde este valor corresponde a 2.

2.-) **Acciones**: Del entorno, el agente selecciona para cada requerimiento R_t un camino entre los K candidatos disponibles y una cantidad de J FSU asignados, es por lo anterior que el espacio de acción (denotado como A) incluye un numero de K*J acciones. La acción seleccionada corresponderá a la que presente el mayor valor dentro de la distribución de probabilidad, y que se verá reflejado en la recompensa de las acciones tomadas.

3.- **Recompensa**: Se obtiene una recompensa inmediata de reward en caso de que R_t es acceptada en donde corresponde a 1 si es aceptado y -1 si no.

4.-**Red Neuronal Densa (DNN)**: DeepRMSA emplea una red neuronal $f_{\theta_p}(s_t)$ para generar la política de RMSA π_t y otra DNN $f_{\theta_v}(s_t)$ para estimar el valor de s_t.Por otra parte, θ_p y θ_v corresponde al set de parámetros de la política y valores de las DNN, $f_{\theta_p}(s_t)$ y $f_{\theta_v}(s_t)$ comparten la misma arquitectura "Fully conected DNN" excepto por la salida en donde $f_{\theta_p}(s_t)$ tiene una salida de K*J y $f_{\theta_v}(s_t)$ solo tiene una neurona de salida.

5.1.3.2. Entrenamiento

Esta etapa corresponderá a un agente de DRL será entrenado bajo tres escenarios de entrenamiento distintos, los cuales serán comparados con respecto al rendimiento de heurísticas de restauración dada por KSP-FF y del esquema de protecciones dedicadas 1+1 utilizada usadas en las EON al momento de su evaluación. Para el caso del entrenamiento corresponde a una de las ideas centrales del trabajo ya que se presentarán fallas dentro del entorno de operación del agente, con lo cual el agente se ve forzado a explorar alternativas que puedan disminuir los impactos de los cambios en la EON, correspondiendo a una novedad en el entrenamiento de agentes DRL. Estos escenarios corresponden a:

1. **Agente Sin Fallas (ASF)**: En donde la política de entrenamiento de este agente es trabajar en una EON dinámica sin fallas y obtener la convergencia en su estado de operación normal.

2. **Agente con una Falla (A1F)**: En este escenario de entrenamiento, al agente se le generará una falla de un enlace en la red (correspondiente a la saturación de los recursos espectrales del enlace), este procedimiento se repetirá por cada enlace de la red, agregando un entrenamiento adicional correspondiente a la operación normal de la EON.

3. **Agente con tres Fallas (A3F):** Al igual que el caso de A1F, este agente se le incorporará una política de tres fallas las cuales transcurrirán en su proceso de entrenamiento en distinto instante de tiempo, distintos enlaces y distintas duraciones de cada falla. Además, ninguna falla se superpondrá con otra, por lo que cada vez que exista una nueva falla, la falla anterior ya ha sido restaurada.

El entrenamiento corresponderá a una de las partes fundamentales del problema que se quiere resolver, debido a que le otorgará la cognición al agente para actuar en las evaluaciones posteriores que se realizarán. Cabe destacar, que dado que forzamos que el agente encuentre rutas alternativas, para el caso de los escenarios con fallas, la métrica que nos indicará si el procedimiento de aprendizaje fue el correcto estará dado por la probabilidad de bloqueo (BP) entregada en las ecuaciones (3) y (4)

Finalmente, el entrenamiento presentará un total de 800.000 iteraciones por escenario, ya que nos asegurará una convergencia en los estados de operación de la red y como el agente interactúa. Los parámetros de los algoritmos utilizados por los agentes corresponderán a las otorgadas por las librerías de Python y expuesta en la Tabla 1.

Elemento	**Valor**
FSU a acomodar en la simulación	320
Demanda de tráfico al azar	[25-100]Gbps
Hiperparametros red neuronal	Aprendizaje: Y=0,95; α=0,01
	Experiencia: N=50; ε_o=0,05; ε_{min}=0,05
Tazas	Mean holding($1/\mu$)=7,5[UT]

	Mean Interraival ($1/\lambda$) =1/12 [UT]
Trafico total (λ/μ)	90 Earlangs

Tabla 1: Parámetros para las simulaciones.

5.2. Reinforcement learning

Dentro de las aplicaciones de ML existe una ramificación la cual corresponde al RL, la cual es ampliamente usado y estudiada para los sistemas de control para optimizar decisiones. Se basa en el proceso "Trial and Error" el cual consiste en un balance en cuanto a las decisiones de explotación o a las decisiones de exploración del agente visualizando el entorno. De lo anterior, el agente va aprendiendo las decisiones optimas interactuando con el ambiente[31].

Para tener una idea de cómo funciona el RL, en [48] se caracterizaron los parámetros de RL los cuales se encuentran en la Tabla 2.

Parámetros	Definición
$\mathbf{T}$	Corresponde al set de tiempos de iteración, incluyendo la secuencia discreta de pasos de iteración t, cada ciclo de iteración el agente completa una acción con el entorno.
$\mathbf{S}$	Usado normalmente como S_t indica el entorno o estado donde se va desarrollando el agente.
$\mathbf{A}$	Indica la acción que toma el agente dado el tiempo donde se desarrolla el entorno, su nomenclatura es a_t o A_t.
$\mathbf{P_t(S_{t+1}\|S_t, a_t)}$	Corresponde a la probabilidad de transición de pasar del entorno S_t a S_{t+1} bajo la acción a_t.
$\boldsymbol{\pi}$	Es la política que mapea el entorno S_t y la acción A_t, por lo que $\pi(s,a)$ nos indica la probabilidad de que el agente actúe bajo la acción a en el entorno s.
$\boldsymbol{r_t(S_t, a_t)}$	Indica el retorno de la acción a_t del agente en el entorno s_t.(reward)

Tabla 2: Principales parámetros del Reinforcement learning.

En general, los algoritmos de RL tienen el siguiente funcionamiento a partir de los parámetros entregados en la Tabla 2. El agente observa el entorno s_t y elige una acción a_t bajo el principio que maximice el retorno $r_t(S_t, a_t)$. A continuación, el entorno transfiere el estado s_t a s_{t+1} terminando de esta manera el ciclo iterativo del agente bajo la acción a_t. Lo anterior se repite un gran número de veces para que el agente se entrene y encuentre los máximos valores de retorno. Lo anterior se visualiza en la Figura N°7.

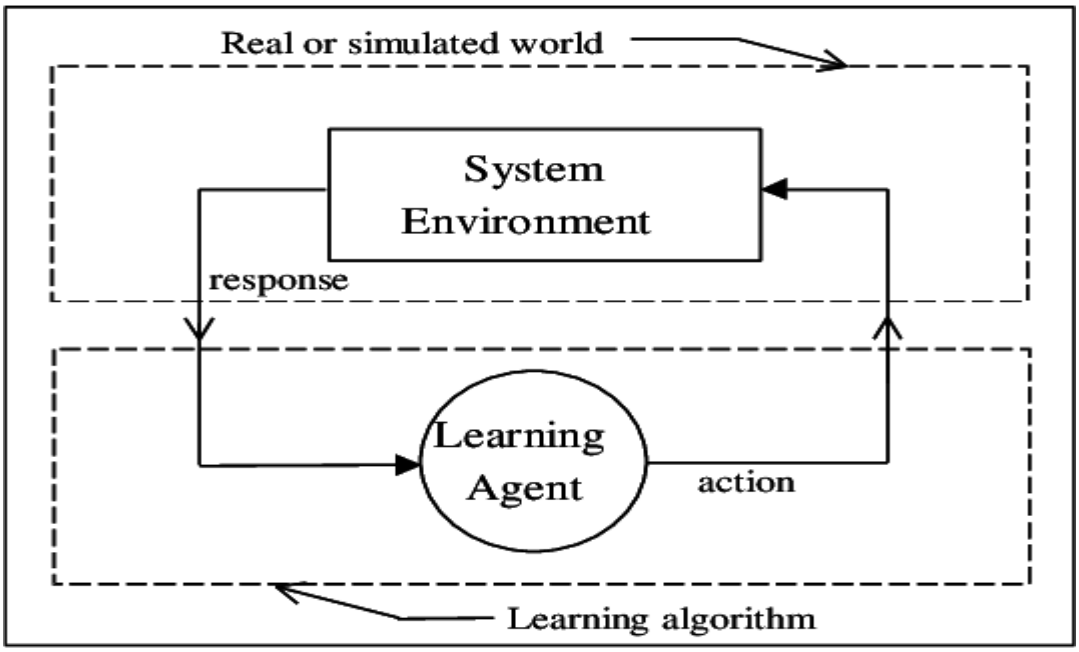

Figura N°7: Esquema general de Reinforcement Learning [49]**.**

Una gran mayoría de los problemas de RL son descritos por "Markov Decision Process (MDP)", es decir que los agentes de RL satisfacen las propiedades de Markov[50], estas propiedades están definidas como:

$$P[S_{t+1}|S_t] = P[S_{t+1}|s_1, \dots, s_t] \quad \textbf{(6)}$$

Las variables S_t y S_{t+1} corresponden a el entorno actual y al próximo entorno del agente respectivamente, en donde t corresponde a un tiempo discreto, la ecuación (6) indica que el agente en el estado S_{t+1}, solo va a depender del estado anterior S_t.

Se puede definir el MDP como un conjunto de estados y acciones, en donde "State space" engloba todos los posibles estados del agente en el tiempo t. Lo anterior se expresar como S_t, y todas las posibilidades del entorno quedan definidos por $s_t \in S_t$, por otra parte, todo el set de acciones que el agente puede tener queda dado por $A(t)$en el tiempo t, en donde $a_t \in A_t$ es la acción tomada por el agente.

La probabilidad del próximo estado y recompensa $(s_{t+1},\ r)$,está dada por los actuales pares de entorno o acción ("state-action pair") quedando expresada como:

$$P(S_{t+1}, r|s_t, a) = \Pr\{S_{t+1} = s', r_{t+1} = r|\, S_t = s, A_t = a\} \quad \textbf{(7)}$$

De la ecuación (7) se pueden evaluar y tener en cuenta 3 caracteristicas, las cuales corresponden a como el "expected reward for a state-action pair", "state-transition probabilities" y "expected reward for a state-action-next-state" [51] y serán a continuación.

5.2.1. Recompensa Esperada.

La recompensa(reward) corresponde al valor que el agente obtiene al momento de tomar alguna acción en el entorno donde está el agente, lo anterior se visualiza en la ecuación (8).

$$r(s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] \quad \textbf{(8)}$$

El agente siempre trata de maximizar el retorno obtenido del entorno, el cual corresponderá la suma de todos los retornos.

$$R_t = r_t + r_{t+1} + r_{t+2} + \cdots \quad \textbf{(9)}$$

De la ecuación (9), se obtiene que si se suman todos los retornos estos pueden llegar a sumar infinito, lo cual generaría un loop que no terminaría las iteraciones. Por lo anterior, se

agrega un parámetro dado por γ que corresponde al factor de descuento. En donde, $\gamma \epsilon [0,1[$ el cual nos indicará la importancia de los futuros retornos, con la consideración que si $\gamma = 0$ solo se toma el retorno inmediato. Por lo tanto, redefiniendo la ecuación (9) queda la expresión:

$$R(t) = \sum_{k=0}^{\infty} \boldsymbol{\gamma}^k r_{t+k+1}\,(S_{t+k}\,, a_{t+k}) \qquad \textbf{(10)}$$

El principal objetivo de las iteraciones consiste en encontrar la política optima π^*que nos permita maximizar el retorno. Estas evaluaciones de políticas estarán dadas por las métricas de "state-value function" dado por $V^{\pi}(S)$ y "action value function" dado por $Q^{\pi}(s,a)$.

5.2.2. Probabilidad de transición de estados (State-transition probabilities).

Corresponde a la probabilidad de encontrar los posibles próximos entornos y retornos, dado por el entorno y la acción actual que tiene el agente, lo anterior está definido como:

$$P(s_{t+1}|s_t, a_t) = \Pr\{s_{t+1} = s'|s_t = s, a_t = a\} \qquad \textbf{(11)}$$

5.2.3. Recompensa esperada por próximo estado-acción (Expected reward for a state-action-next-state).

Como se mencionó en la sección Recompensa Esperada.5.2.1. El principal objetivo de las iteraciones consiste en encontrar la política optima π^*que nos permita maximizar el retorno, estas evaluaciones de políticas estarán dadas por las métricas de "state-value function" dado por $V^{\pi}(S)$ y "action value function" determinada por $Q^{\pi}(s,a)$.

Para el caso de "sate-value-function" está definida por la ecuación (12) y nos indica el retorno esperado cuando se parte bajo el entorno S seguido por la política π.

$$V^{\pi}(S) = \mathrm{E}_{\pi}[R_t|S_t = s] = \mathrm{E}_{\pi}[\sum_{k=0}^{\infty} \boldsymbol{\gamma}^k r_{t+k+1}\,(S_{t+k}\,, a_{t+k})|\, S_t = S] \qquad \textbf{(12)}$$

Por otra parte, el "action-value-function" $Q^{\pi}(s,a)$ es el valor esperado tomando la acción de salida, bajo la política π, lo anterior está definida por la ecuación (13).

$$Q^{\pi}(S,a) = \mathrm{E}_{\pi}[\sum_{k=0}^{\infty} \boldsymbol{\gamma}^{k} r_{t+k+1}\,(S_{t+k}\,, a_{t+k})|\, S_t = S, a_t = a] \qquad \textbf{(13)}$$

Teniendo en cuenta que el agente RL trata de buscar una política óptima, se tiene que encontrar alguna manera que comprare estas dos políticas. Para esto, se define que una política π es mejor o igual a la política π' si y solo si el retorno obtenido es mayor o igual que el retorno de la política π', lo anterior se puede escribir como:

$$\pi \geq \pi' \leftrightarrow v_{\pi}(s) \geq v_{\pi'}(s) \qquad \textbf{(14)}$$

La ecuación (14), nos indica que la política óptima que toma el agente siempre estará entregada cuando el proceso de entrenamiento converja a algún valor. Por lo tanto, cuando se encuentra la política optima π^{*}, los valores de $V^{\pi}(S)$ y $Q^{\pi}(S,a)$ son maximizado, reescribiendo las ecuaciones (12) y (13) de tal manera que indiquen la política optima a la que se desea llegar, lo anterior queda:

$$\pi^{*} \begin{cases} V^{*}(S) = argmax_{\pi} V^{\pi}(S) \\ Q^{*}(S,a) = argmax_{\pi}\, Q^{\pi}(S,a) \end{cases} \qquad \textbf{(15)}$$

Para aplicar el RL dentro de nuestro problema, se debe seleccionar algoritmos que lo permitan trabajar y optimizar minimizando la probabilidad de bloqueo, es decir, necesitamos seleccionar un agente que sea capaz de obtener los mejores retornos en las evaluaciones después que este agente es entrenado, de lo anterior existen varios algoritmos, tales como "Deep Q-learning" y "Advantage Actor Critic (A2C)".

5.2.3.1. Deep-Q-Learning

Tomando en cuenta la definición de "action-value-function" $Q^{\pi}(s,a)$, nos indica que retorno esperado bajo el "state-action pair". Modificando la ecuación (13) y aplicando la ecuación de Bellman se obtiene:

$$Q^{*}(s,a) = \mathbb{E}[r + \gamma \max_{a'} Q^{*}(s',a')] \quad \textbf{(16)}$$

La expresión (16), nos indica que el máximo retorno de los entornos y acciones (s,a) es la suma de los retornos anteriores y el factor de descuento bajo la política tomada. Iterando la ecuación (16) se obtiene:

$$Q_{t+1}(s,a) = \mathbb{E}[r + \ \gamma[r + \gamma \max_{a'} Q_t(s',a')]] \quad \textbf{(17)}$$

La ecuación (17) va a converger en el óptimo de la función Q, dada por la política de la expresión (14), es decir:

$$\lim_{t\to\infty} Q_t = Q^{*} \quad \textbf{(18)}$$

Cuando se tiene un gran número de pares entorno-acción y la estimación separada de la función action-value, no es practico realizar una siguiente iteración para obtener el óptimo action-value que se quiere trabajar. De lo anterior, para el caso del Deep Q leaning usa una red neuronal como función de aproximación con los parámetros θ dada por $Q(s,a;\theta) \cong Q^{*}(s,a)$. Este algoritmo es entrenado cambiando los valores de θ_t en cada tiempo para reducir el error entre la entrada y salida de la función Q, para realizar lo anterior se trata de minimizar cada iteración t mediante la ecuación:

$$Loss(\boldsymbol{\theta}_t) = \mathbb{E}\{[(r + \gamma\, max_{a'}\, Q(s',a')) \ - Q(s,a,\theta_t)]^2\} \quad \textbf{(19)}$$

De la ecuación (19),la expresión $(r + \gamma \max_{a'} Q(s', a'))$ corresponde al valor esperado mientras que $Q(s, a, \theta_t)$ corresponderá al valor obtenido en el tiempo t, entregando la diferencia temporal del error obtenido, esta diferencia dada por el error actualiza los parámetros θ de la red neuronal.

La manera de trabajar del algoritmo Q leanrnig es descrita como una “off-policy” lo que indica que determina la política óptima independiente de las acciones del agente, para esto utiliza la estrategia “$\epsilon - greedy$" la cual nos permite elegir una acción “greedy” de probabilidad $1 - \epsilon$ y una acción random de probabilidad ϵ[52].

5.2.3.2. Advantage Actor critic (A2C)

Este algoritmo utiliza dos redes neuronales en paralelo, un actor y un crítico. El actor modela la política de la función que controla como el agente trabajará bajo las acciones disponibles, lo cual puede ser definido como $\pi(s, a, \theta)$. Por otra parte, el crítico corresponde a “value-function” que revisa las acciones tomadas por el agente en cada tiempo t, esto es representado como $\hat{q}(s, a, w)$. De lo anterior, el actor va actualizando los parámetros de su política para las futuras acciones utilizando el feedback del crítico, de esta manera complementándose en su manera de actuar[53].

Este método utiliza la siguiente función $A(s, a) = Q(s, a) - V(s)$, en donde $Q(s, a)$ corresponde el valor de la acción a en el entorno s y $V(s)$ corresponde a la media valor de ese entorno. Lo anterior se escribe de esta manera para disminuir los escenarios en los cuales se presente una gran variabilidad.

5.3. Entorno de Simulación “Deep S-RMSA”.

En esta sección se expondrá sobre el modelo modificado e implementado para el desarrollo de los principales objetivos del trabajo. Por otra parte, se presentará los métodos de entrenamiento utilizado y la reacción del agente ante los distintos entornos de entrenamiento.

Finalmente, se dan a conocer los resultados y comparaciones entre los agentes y heurísticas aplicadas para el desarrollo del trabajo.

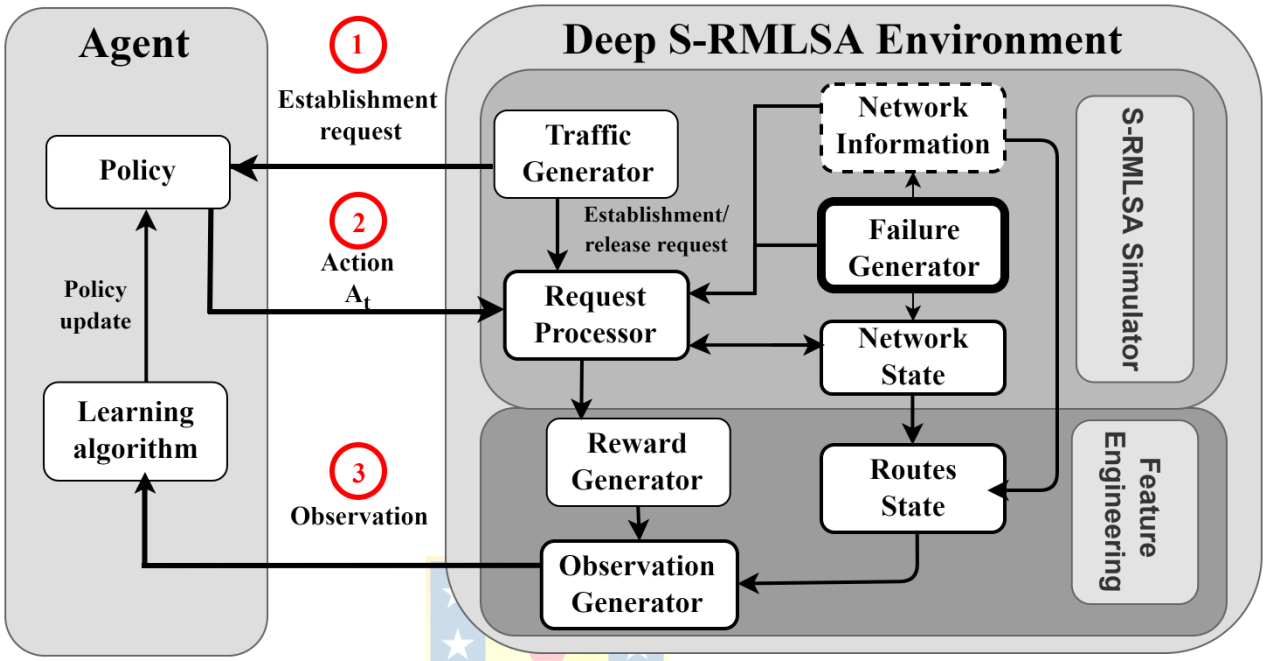

Figura N°8: Esquema del Nuevo Entorno DeepSRMLSA y su Implementación de Fallas.

La Figura N°8, nos indica el entorno creado para poder generar las fallas en el entorno OpenAI gym de Natalino [37]. Se pueden visualizar los distintos módulos con el que se ejecuta, este entorno presenta las mismas caracteristicas entregadas por [25], pero en este caso se creó el módulo Generado de Fallas (Failure Generator), el cual tiene la capacidad de elegir un enlace e impedir que este pueda o transportar cualquier conexión en un tiempo determinado o ilimitado dentro de ese enlace. El modelamiento de la falla corresponde a $F_{sl,dl} = 0$, en donde F corresponde a los slots disponibles entre los nodos de fuente sl y el nodo de destino dl. El agente recibe una constante actualización de los módulos de los estados de la red y de las informaciones de la red y con esa información actualiza los nuevos requerimientos de los usuarios. Por otra parte, la Tabla 3 indica los alcances máximos por cada formato de modulación aplicado en el entorno, la cual limita las opciones del agente ya que en este caso deberá tener en cuenta que formato utilizar según el camino que seleccione.

La selección de las rutas del trabajo estará dada por el algoritmo KSP, y se pre calcularan de la siguiente manera. Al agente se le asignarán 5 rutas, de las cuales dos rutas pasarán a ser disjuntas y las rutas restantes compartirán enlaces con una de las rutas disjuntas, es decir, que las rutas [1,2,3] compartirán enlaces, mientras que el caso de las rutas [4, 5] corresponderán a rutas disjuntas de las primeras 3 rutas, pero que entre ambas rutas compartirán por lo menos un enlace. Lo anterior nos entrega distintas rutas que dispondrá el agente al momento de tomar la decisión sobre cual acción tomar.

Formato de Modulación	Maximo Reach (km)
32QAM	500
16QAM	1.000
8QAM	2.000
QPSK	4.000
BPSK	8.000

Tabla 3: Máximo Alcance según el Formato de Modulación[54]

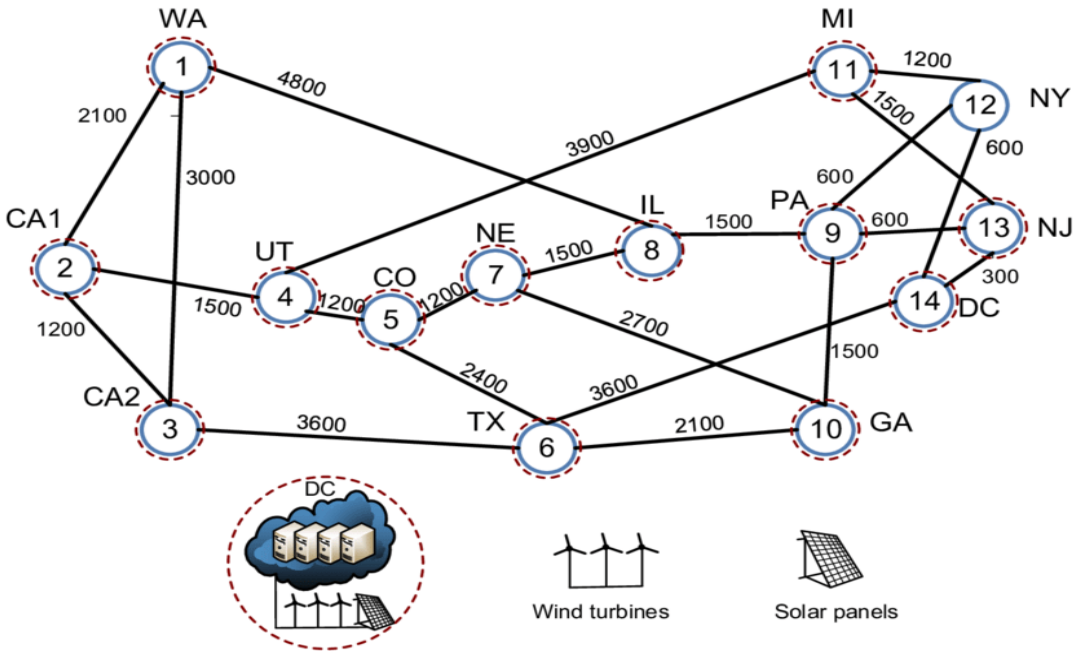

Figura N°9: Topología de la NSFNET[55].

En la Figura N°9 se expone la topología de la red NSFNET, con la que trabajará y además se enumeran los enlaces según los nodos de fuentes y destinos en la Tabla 4.

Enlace	Nodo Inicial	Nodo Final
0	1	2
1	1	3
2	1	8
3	2	3
4	2	4
5	3	6
6	4	5
7	4	11
8	5	6
9	5	7
10	6	10
11	6	14
12	7	8
13	7	10
14	8	9
15	9	10
16	9	12
17	9	13
18	11	12
19	11	13
20	12	14
21	13	14

Tabla 4:Enumeración de Enlaces y nodos de la Red NSFNET.

Dada la cantidad de enlaces y nodos que presenta la red, no todos los enlaces tienen la misma importancia en caso de falla, es por lo anterior que se necesita identificar cuales corresponderán a enlaces donde pasen una mayor cantidad de rutas al momento de la aplicación del algoritmo KSP y en cuáles no. En caso de que un enlace pase muchas rutas de conexión entregada por la heurística se le denominará enlace central, mientras que en el caso contrario se denomina enlace periférico o de borde. Para obtener estos resultados, se aplica la centralidad de intermediación de un grafo la que nos entrega, cual es la frecuencia que un nodo se encuentre entre los caminos más cortos entre nodos distantes[56].

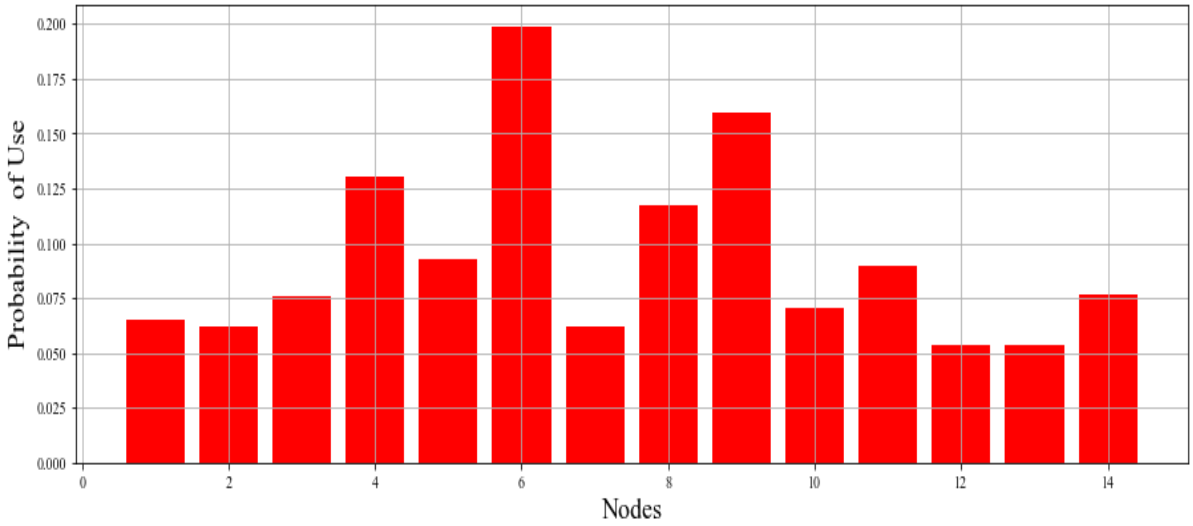

Figura N°10: Probabilidad de uso de un nodo por los caminos más cortos utilizados.

En la Figura N°10 se observa que los nodos 4, 6, 8 y 9 son los que presentan mayor probabilidad de uso, por lo que si se genera una falla de los enlaces que bordea estos nodos, generarán un alto impacto en la red. En este trabajo se define como enlaces centrales los enlaces 4, 6, 8, 11, 14, 15 y 17 de la Tabla 4, mientras que los otros enlaces corresponden a enlaces periféricos.

6. Resultados.

6.1. Entrenamiento

El entorno de entrenamiento se realizará con las siguientes caracteristicas. Los formatos de modulación utilizados son BPSK, QPSK, 8-QAM y 16-QAM con sus alcances definidos en la Tabla 3. La capacidad de cada enlace corresponderá a 100 FSU, en donde cada FSU presenta un ancho de banda de 12,5GHz. Se asume un sistema dinámico de solicitudes (conexiones), donde las solicitudes de establecimiento de conexión están basadas en un proceso de Poisson y los tiempos de espera de la conexión siguen una distribución exponencial negativa. El Bit-Rate está dentro de en un rango de [25-100]Gbps. El agente seleccionará una de las 5 rutas pre calculadas en donde al menos dos de rutas son disjuntas por lo que siempre el agente presenta caminos alternativos. El agente se entrenará en episodios de 50 solicitudes de conexión cada uno (para simplificar la backpropagation en la DNN utilizada por el agente al entregar pequeños lotes de datos de forma continua), y el número total de pasos en el entrenamiento será de 800 000.

La DNN consta de 5 capas de neuronas totalmente conectadas entre sí y con las funciones de activación *elu*. La entrada corresponde al modelamiento expuesto en 5.1.3.1 y la salida corresponderá a las K*J, donde K indica la cantidad de caminos pre calculados y J los números de slot asignados.

Para el entrenamiento se utiliza un agente A2C en tres entornos simulados. Nos referiremos a estos tres entornos como distintos agentes, en donde el primer agente es entrenado sin ninguna falla en sus enlaces, por lo tanto, opera en un estado dinámico normal. El segundo agente, fue entrenado en un entorno en donde se le genera una falla de un enlace por un tiempo determinado, este proceso es repetido por todos los enlaces de la red y se repite una vez más considerando el estado de operación normal del sistema (sin falla). Finalmente, el tercer y último agente corresponde al agente entrenado con tres fallas, de las cuales dos fallas corresponderán a fallas de los enlaces de los bordes y una falla corresponderá a un enlace central de la red. Todos, estos agentes serán presentados en las secciones posteriores, cabe

destacar que el hecho que le agreguemos fallas a los agentes en el proceso de entrenamiento forzará el espacio de acción basado en KSP-FF del agente y deberá presentar una exploración de su entorno, encontrando y seleccionando las mejores rutas para enviar las conexiones requeridas, como también disminuyendo la probabilidad de bloqueo de la red.

Dentro del proceso de entrenamiento, cuando el agente asigna alguna conexión dentro del espectro óptico de una ruta preseleccionada, más conexiones podrán ser procesadas y asignadas, de esta manera la política de acción es mejorada. Por otra parte, cuando el proceso de entrenamiento termina, la política del agente será tal de elegir las mejores rutas según la mayor probabilidad de que estas rutas no estén bloqueadas, aun cuando exista alguna falla dentro del proceso y mediante esto enviar la información solicitada por los usuarios.

6.1.1. Agente Sin Fallas (ASF)

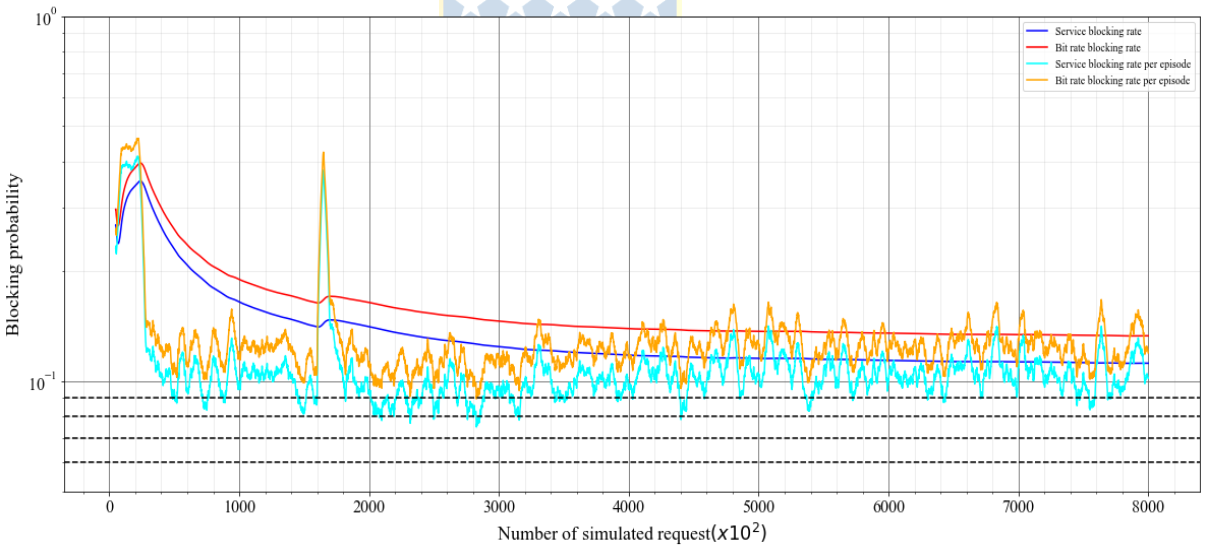

Figura N°11: Entrenamiento Agente SF A2C.

En la Figura N°11 se presenta el entrenamiento del agente ASF de DRL con un algoritmo de aprendizaje de A2C, de StableBaseline[57]. Este proceso de entrenamiento solo se repite una vez. Se aprecia un estado de operación normal de la red en donde las primeras iteraciones del entrenamiento. El agente presenta una alta probabilidad de bloqueo ya que son los primeros pasos antes de aprender las nociones de como asignar el espectro óptico dentro de

las rutas preseleccionadas. Por lo tanto, esta sección corresponde a la exploración del agente en la red. Entre las iteraciones 100.000 y 200.000, se aprecia un aumento en la probabilidad de bloqueo dada por la saturación de la red y la imposibilidad del agente de poder posicionar el espectro óptico de las conexiones solicitadas en esos instantes. Este peak presenta una corta duración y por el resto del tiempo del entrenamiento el agente presenta un estado estacionario con oscilaciones dadas por las distintas cargas de tráfico entregadas en un estado dinámico. Estos valores convergen a una probabilidad de bloqueo en torno a los 0.1.

En caso de que una conexión no se pueda asignar, el agente lo tomará como una recompensa negativa y corresponderá a un aumento de la probabilidad de bloqueo.

6.1.2. Agente con 1 Falla (A1F)

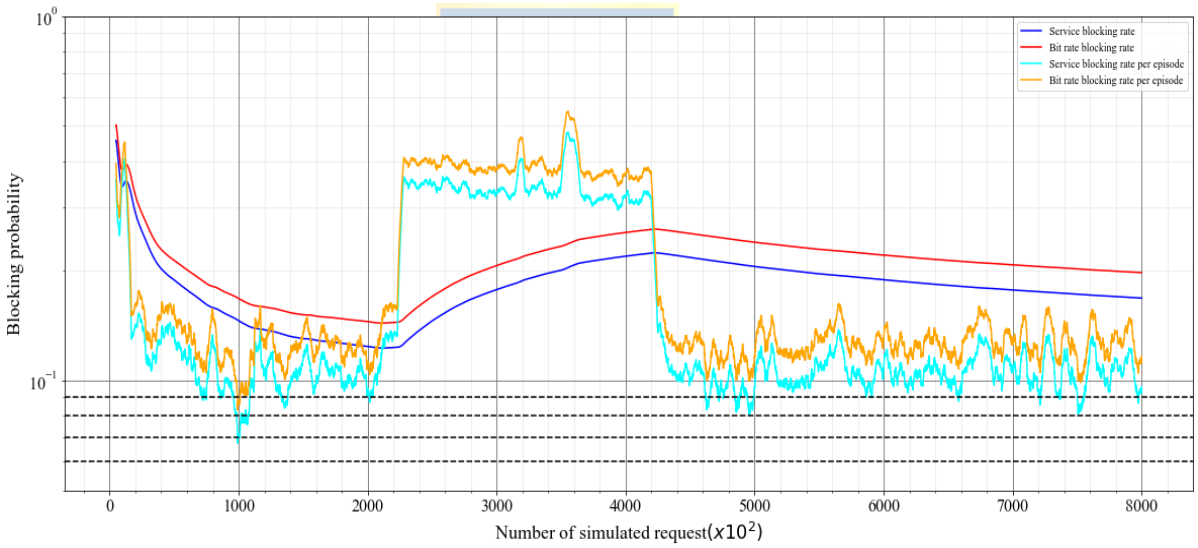

Figura N°12: Entrenamiento Agente 1F A2C.

En la Figura N°12, se presenta uno de los entrenamientos del agente A1F, este caso corresponde a uno de los últimos entrenamientos realizados al agente. Se genera una falla en el enlace 14 de la topología NSFNET (Figura N°9), esta falla corresponde a un enlace central de la red óptica, por lo cual presenta un alto impacto en la probabilidad de bloqueo al momento de generar la falla. En este caso, al igual que el agente ASF, el agente A1F presenta una alta probabilidad de bloqueo al inicio del entrenamiento dado que el agente está empezando la asignación del espectro óptico de las conexiones requeridas, además existe una diferencia importante con

respecto a la duración temporal del inicio del bloqueo, en donde esta diferencia expresa el conocimiento adquirido hasta el momento por A1F. Por otra parte, la falla generada por el enlace central de la red, además de presentar una alta probabilidad de bloqueo, la falla presenta una larga duración, la cual obliga al agente a tomar acciones aleatorias la cuales puedan aumentar o minimizar su recompensa, estas acciones aleatorias se pueden visualizar en los dos peaks generados durante la falla, en donde el agente toma acciones para poder encontrar caminos por donde enviar la información requerida, pero estas acciones fueron infructuosa lo cual produce que el agente reciba un minimización en su recompensa y vuelva a su estado previo.

Una vez liberadas la falla, el agente regresa rápidamente a su estado original. Los anteriores resultados, nos indican dos grandes conclusiones, la red ante fallas en enlaces centrales genera un aumento significativo de la probabilidad de bloqueo, dificultando y saturando el tráfico de la red, por otra parte, la red óptica es susceptible para cualquier cambio en el estado dinámico de la red, por lo tanto, el agente controlador SDN debe ser capaz de manejar y reestablecer las rutas utilizadas previas a la falla y llevar al estado de operación dinámico previo a la falla, esto siempre y cuando la cantidad de trafico de la red sea estable.

Cabe destacar, que las duraciones de las fallas en todos los procesos de entrenamiento presentan la misma duración, de esta manera se deja en igualdad de condiciones para que el agente explore las fallas centrales y las fallas de borde.

6.1.3. Agente con 3 Fallas (A3F)

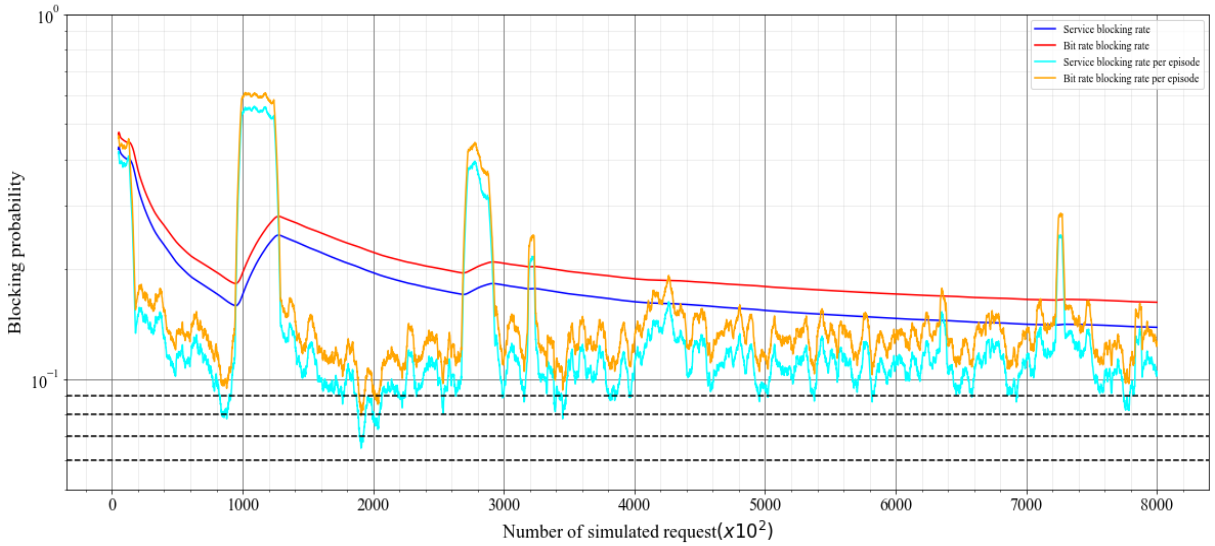

Figura N°13:Entrenamiento Agente 1F A2C.

En la Figura N°13, se presenta uno de los procesos de entrenamiento del agente A3F, este proceso de entrenamiento corresponde a uno de los últimos entrenamientos realizados por el agente y en el cual se presenta tres fallas en los enlaces 14,10 y 0 respectivamente las cuales se pueden visualizar en la Tabla 4. Al igual que en el caso ASF y A1F, el agente A3F presenta un aumento en la probabilidad de bloqueo al inicio del entrenamiento, la cual cae rápidamente dado el aprendizaje adquirido por el agente. Para el caso de la primera falla corresponde a una falla de un enlace central de la red, pero la diferencia está dada por el tiempo de duración con respecto al agente A1F, la cual en el caso del agente A3F fue de un menor tiempo. Lo anterior, fue dado ya que se planea que el agente presente distintos procesos de exploración comparando sus resultados. Por otra parte, la segunda falla correspondiente a una falla de borde de la red, presenta un comportamiento en particular en el cual mediante el proceso de exploración el agente disminuye la BP en uno de sus peaks, esto se debe a una acción particular del agente, el cual selecciona uno de los KSP disponibles, lo que nos indica que el agente presenta un conocimiento de lo que debe realizar. Por último, la tercera falla generada corresponde a un enlace de borde. Se observa una de las caracteristicas más importantes de los procesos de Reinforcement learning, correspondiente al aprendizaje del agente, ya que esta falla, la probabilidad de bloqueo es casi imperceptible, lo que nos indica que el agente sabe que decisiones tomar, cuando se produce una falla dentro de los enlaces de bordes, haciendo casi imperceptible la falla generada.

Liberadas las fallas mencionadas anteriormente, la probabilidad de bloqueo disminuye hasta la operación dinámica normal como en los casos anterior de los agentes, lo cual nos indica que todos los agentes presentados tienen la capacidad de rápida adaptación a los cambios de entornos. Por otra parte, este proceso fue repetido 25 veces tratando de obtener todas las posibles combinaciones que se pueden generar.

6.1.4. Recompensa del Entrenamiento de los Agentes.

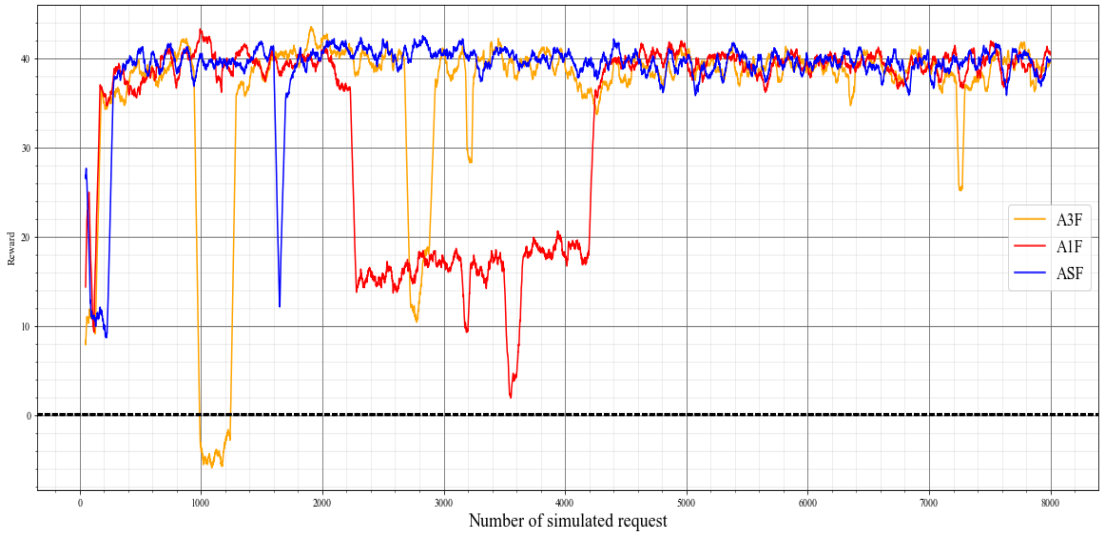

Figura N°14: Recompensa de los agentes en el entrenamiento A2C.

Para el caso de la recompensa de los agentes, se trabajó con los mismos parameros presentados por el trabajo de [37] Los valores de la recompensa estarán dado por 1 y -1, en donde 1, es cuando el agente puede entregar la conexión solicitada por el usuario y -1 en caso rechazo de conexión, esto se puede dar por la imposibilidad del agente de asignar el recursos por las saturación del enlace, que se supere el alcance máximo del reach dada el formato de modulación aplicado y/o fallas de enlace.

En la Figura N°14, se visualiza la recompensa obtenida por los agentes en el proceso de entrenamiento para el algoritmo A2C. En este caso todos los procesos de bloqueos aplicados

en la EON son reflejados como recompensa negativa disminuyendo la recompensa en el proceso de iteración, lo anterior se verifica con el comportamiento de los agentes A1F y A3F. Una vez que todos los agentes terminan sus estados de falla, las recompensas que se obtienen convergen dentro del mismo punto, que en este caso corresponde a 40. Esto nos da a entender que los agentes, obtienen un aprendizaje similar para establecer las conexiones necesarias, pero con distintos entornos. Existe un caso particular, para la recompensa del agente A3F, cuando se le genera una falla en un enlace central, ya que la recompensa es negativa, eso nos indica que al inicio de la falla todas las conexiones que se solicitaron fueron bloqueadas, dado que se genera un cambio repentino en la red.

6.2. Evaluación

Para el entorno de evaluación, se ocupa la gran mayoría de las caracteristicas presentadas en el entorno de entrenamiento, pero los cambios y principales parámetros utilizados serán mostrados en la Tabla 5.

Parámetros	Valor
Parámetros Red	
Topología	NSFNET
Numero de FSU por enlace	100 FSU
Formatos de modulación	BPSK, QPSK, 8-QAM y 16-QAM
Parámetros Trafico	
Bit-Rate[Gb/s]	Distribución uniforme [25-100]Gbps
Parámetros de evaluación del Agente	
Rutas Pre-Calculadas	5
Conexiones requeridas por episodio	50
Simulaciones requeridas para la evaluación	10.000
Parámetros del algoritmo de aprendizaje del agente	Por default

Tabla 5: Características de evaluación del entorno del agente.

La evaluación que se le debe realizar a los agentes debe ser con una heurística con la cual se pueda comparar dentro de las mismas métricas y además que le heurística cumpla la función de resolver el problema de S-RMLSA, de esta manera se pueda igualar la mayor cantidad de condiciones entre los agentes y la heurística. Es por lo anterior que se decidió utilizar el esquema de protección dedicada 1+1 [58] y la heurística de restauración KSP-FF dado por el espacio de acción de nuestro agente [25]. El esquema 1+1 de protecciones dedicadas tiene la característica de que selecciona dos rutas disjuntas las cuales son, ruta de

trabajo (WP) en la cual se calcula el número de slot necesarios para enviar las conexiones solicitadas por el usuario y una ruta de protección (PP), la cual se activa en caso de que la WP no pueda enviar la conexión o exista una falla y los servicios que se siguen solicitando se vean interrumpidos. Es por esto que la ruta del PP es disjunta a la WP, lo que hace calcular de forma paralela el número de slot necesarios para el PP y guardar espectro óptico de ser necesario el tiempo necesario para poder enviar la información. En caso, que en ninguna de las rutas se pueda enviar la información, ya sea porque el reach necesario supera la distancia establecida o que la capacidad de los enlaces se ven sobrepasados, se considerará como una conexión rechazada. La forma en cómo se escribió el código se visualiza en la Figura N°15.

Require: Physical topology G = (E,V) and connection demands $C = \{u_c, v_c\}$;
1: Find working paths using Dijkstra's algorithm;
2: Apply possible FS;
3: Find backup paths for all connection demands using Dijkstra's algorithm;
4: τ = set of working paths;
5: σ = set of backup paths;
6: $\Phi = V$;
7: **while** τ !=∅ and σ !=∅ **do**
8: Find the node j in Φ traversed by the greatest number of lightpaths from τ and σ;
9: Add one additional BV-WSS in node j;
10: Remove node j from the set of candidate nodes Φ for adding additional redundancy;
11: Remove all working paths which use node j from τ;
12: Remove all backup paths which use node j from σ;
13: **end while**

Figura N°15: Esquema de Protección Dedicada 1+1[58]**.**

Considerando un algoritmo de comparación para evaluar el funcionamiento de los agentes, se definirá la manera en cómo este entorno de evaluación funcionará. Este nuevo entorno trabajará en una situación dinámica normal en los primeros instantes de su implementación, en donde pasadas las 2000 iteraciones se presentará una falla del enlace 14, por lo cual será de alto impacto. La métrica que se utilizará para evaluar el comportamiento de los agentes y heurística corresponderá al bit rate blocking rate.

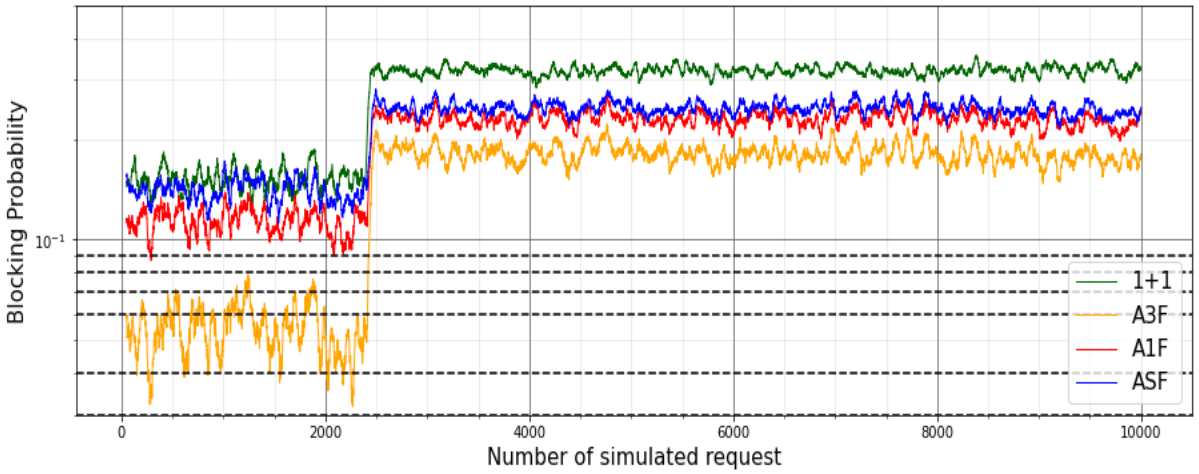

Figura N°16: Evaluación de agente y heurística en la topología NSFNET y el bit-rate Blocking rate obtenido.

En la Figura N°16, se visualiza el proceso de evaluación de los agentes y la heurística, se visualizan 2 estados cruciales: el primero previo a la falla y el segundo estado correspondería a post falla.

Para el estado previo a la falla se observa que el comportamiento de todos los agentes tiene una menor BP que la heurística presentada, estas mejoras están en el orden 7.20[%], 24,25[%] y 64,23[%] para los agentes ASF, A1F y A3F respectivamente. Este comportamiento se debe a que el esquema de protección dedicada 1+1, necesita al menos el doble de espectro necesario para enviar la información de las conexiones solicitadas, debido a que debe utilizar el espectro para el WP y reservar un espectro para el PP. En el caso de los agentes, el proceso anterior no pasa, debido a que ocupan todos los slots disponibles para poder enviar la información requerida, aun así, las diferentes técnicas utilizadas para el entrenamiento de los agentes presentan considerables diferencias en el proceso de evaluación. En primer lugar, el agente ASF, fue uno de los que presento peor métrica, aun cuando este agente se entrenó en un estado de operación dinámica normal, sin embargo, el agente no presentó saturaciones y pudo mantener el comportamiento de la métrica utilizada dentro de un rango. Por otra parte, para el caso del A1F, presenta una mejor distribución de los recursos disponibles, exhibiendo una mejor probabilidad de bloqueo que en el caso del ASF (pero no significativamente mayor) y de la heurística, además no se presenta saturaciones dadas por la asignación de los recursos

manteniendo dentro de un rango la métrica utilizada. Aun así, la diferencia entre el A1F y ASF es pequeña en cuanto a la BP, tomando en cuenta que en el caso del A1F el tiempo de entrenamiento es 22 veces más que en el caso del ASF, el agente A1F no presenta el comportamiento esperado. Finalmente, en el caso del agente A3F, corresponde al agente que presente el mejor rendimiento, superando el rendimiento de la heurística y los agentes, por lo tanto, la capacidad de explotación del aprendizaje obtenido por el agente A3F, se ve reflejado en la forma en como este puede distribuir las conexiones solicitadas, además de mantener la métrica dentro de un rango estable entre los 0.06 y 0.05.

En el caso de post-falla, el comportamiento de la heurística se mantiene con uno de los peores rendimientos, pero con la condición de que el impacto de la falla no genera una saturación total de la red, permitiendo aun transmitir una limitada cantidad de conexiones disponibles. Para el caso del agente ASF, el impacto de la falla genera que el agente presenta un aumento considerable de la probabilidad de bloqueo aumentando cerca de un 74 [%] con respecto a la probabilidad de bloqueo previo a la falla, lo cual genera una gran cantidad de conexiones interrumpidas, al igual que en el caso de previo a la falla, este agente mantiene la probabilidad de bloqueo dentro de un rango estable y disminuye con respecto a la heurística en un 22,79%. Por otra parte, el agente A1F, igual presenta un aumento significativo en su probabilidad de bloqueo la cual aumenta un 100 % aproximadamente con respecto a la BP previo a la falla, pero sigue teniendo una menor BP al agente ASF y al igual que en el caso previo a la falla el agente A1F presenta magnitudes parecidas que agente ASF, disminuyendo la BP con respecto a la heurística en un 28,21%. Lo anterior confirma que el proceso de entrenamiento de este agente presento un comportamiento anormal a lo esperado. Lo anterior, se debe a que cuando un agente presenta anomalías en el proceso de entrenamiento, puede deberse a varios factores, en este caso se presentó un efecto denominado olvido catastrófico [59], este proceso de olvido catastrófico se debió al cómo se entrenó al agente, dado que se repitió el proceso de entrenamiento generando una falla por cada enlace de manera sucesiva y en donde los últimos enlaces del entrenamiento corresponden a fallas relacionadas a los bordes de la red, por lo cual el agente no se vio obligado a aplicar de una manera extrema el proceso de exploración para encontrar rutas alternativas para enviar la información, es por este motivo que la cognición adquirida por el agente corresponderá a una mezcla entre fallas centrales y

fallas de bordes, lo cual explica el por qué presenta una mejora con respecto al BP comparado con el agente ASF.

Finalmente, el agente A3F presenta el mejor comportamiento ante una falla, aunque comparado con el estado del agente previo a la falla, el impacto de la falla hace que la BP aumente en un 229 %, aun así, corresponde al agente que presenta la menor BP disminuyendo en un 43,74 % comparado con la heurística. Al igual que los otros agentes, una vez ocurrida la falla el agente A3F llega a un estado sin saturaciones en donde trabaja dentro de un rango cercano a los 0,2 BP. A diferencia de lo ocurrido con el agente A1F, el agente A3F pudo obtener un aprendizaje tal que existe una diferencia considerable con respecto a la heurística y el agente ASF, esto se debió al tipo de entrenamiento al cual fue sometido el agente A3F dado que en cada iteración del entrenamiento este agente pudo tener una falla central, con otras dos fallas de bordes indicándole los casos más posibles que pueden ocurrir dentro de la red. Por lo tanto, el agente adquirió una cognición más completa de las posibles situaciones, la cual permite una mejor distribución de los recursos disponibles y saber que acciones tomar en casos de fallas de gran impacto como en la Figura N°16. Los datos anteriores se pueden ver resumidos en la Tabla 6, donde en color negro están los valores BP previo a la falla y post falla, y en paréntesis con color azul se presentan la mejora en porcentaje de la BP con respecto a la heurística de los agentes.

Agentes	Previo Falla	Post-Falla
1+1	0,15151	0,311754
ASF	0.1406 (7.20%)	0.24517 (22,79%)
A1F	0.11477 (24,25%)	0.22796 (28,21%)
A3F	0.05419 (64,23%)	0.17865 (43,74%)

Tabla 6: Resumen de Resultados de Evaluación de Heurísticas v/s Agentes Pre y Post Falla.

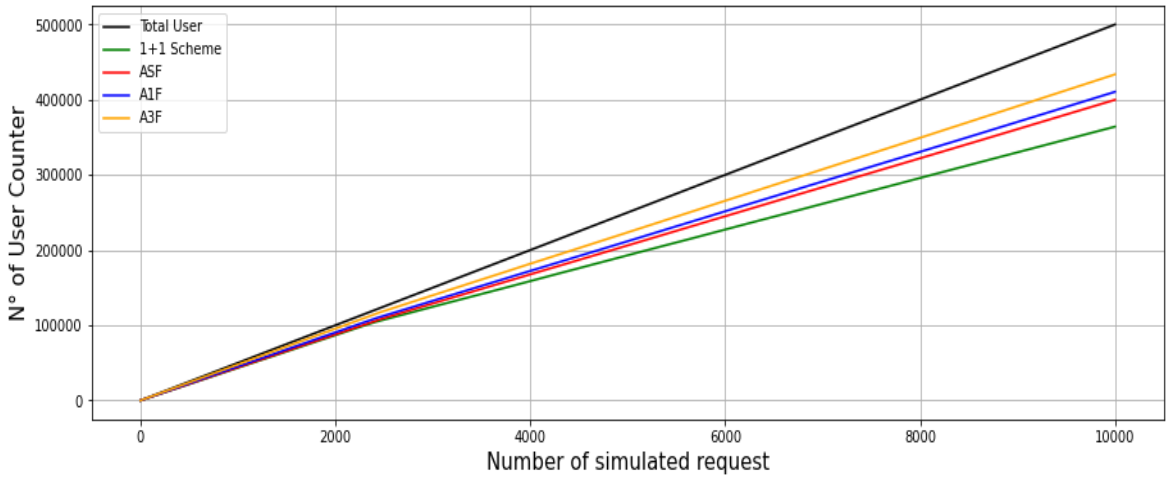

Figura N°17: Número de Usuarios Totales y Usuarios Aceptados.

En la Figura N°17, corresponde a la cantidad de usuarios conectados en función del N° de iteraciones. La línea negra indica la cantidad total de usuarios solicitando servicios, mientras que las otras líneas indican la cantidad de usuarios a los que fueron entregados sus conexiones por la heurística y los agentes. Se puede, confirmar algunos puntos importantes, previo a la falla, la heurística como agentes están entregando un número similar de conexiones solicitadas, por lo cual se hace casi imperceptible las diferencias entre las heurísticas y agentes. Pero una vez que se ejecuta la falla, la situación cambia drásticamente. Al igual que en la Figura N°16 el agente A3F presenta la mayor cantidad de conexiones entregadas teniendo un valor de 433.648 de las 500.000 conexiones solicitadas ratificando como el agente con mejor desempeño y presentando una clara ventaja sobre la heurística que presenta la menor cantidad de conexiones entregadas. Por otra parte, con estos datos se puede obtener una diferencia entre el comportamiento de los agentes A1F y A0F que al momento de visualizarlo con la probabilidad de bloqueo es más difícil de concluir, esta diferencia corresponde a la cantidad de conexiones entregadas, lo cual nos indica la cantidad de clientes que pueden mantener la conexiones post falla, en donde en el caso del agente ASF este valor corresponde 399.815, mientras que en el caso del Agente A1F esta valor corresponde a 410.455 . Lo que nos indica un aumento de capacidad 10.640 conexiones, que en términos comerciales será beneficioso para la compañía de telecomunicaciones. Dándonos a entender que por más que el agente A1F, presentará un olvido catastrófico, aun presenta una ventaja comparativa real sobre el agente ASF, estas comparaciones se ven reflejadas en la Tabla 7.

Agente / Heurística	Número de Usuarios	[%]
Estado Normal	500.000	100
Esquema 1+1	364.119	72,82
ASF	399.815	79,96
A1F	410.455	82,09
A3F	433648	86,73

Tabla 7: Comparación de Números de usuarios Transmitidos v/s Total de la Heuristica y de los Agentes

De la Tabla 7, se obtiene que la idea inicial de entrenar agentes con fallas para poder restaurar los servicios interrumpidos es viable y que aumenta la capacidad de administración de los agentes SDN considerablemente aun en fallas. Lo anteriormente mencionados resultará, siempre que se tenga precaución con los efectos del olvido catastrófico.

Con los resultados anteriores, viene la siguiente inquietud que pasará, si en vez de entrenar toda una red, se enfocará en entrenar el agente solo en las falla de enlaces con mayor impacto en la red y así mejorar las métricas de probabilidad de bloqueo, aumentar la capacidad de conexiones que se pueden entregar y disminuir considerablemente la cantidad de entrenamientos realizados pasando de 23 entrenamientos a solo 5 para el caso del agente A1F. Lo anterior se puede visualizar en la Figura N°18.

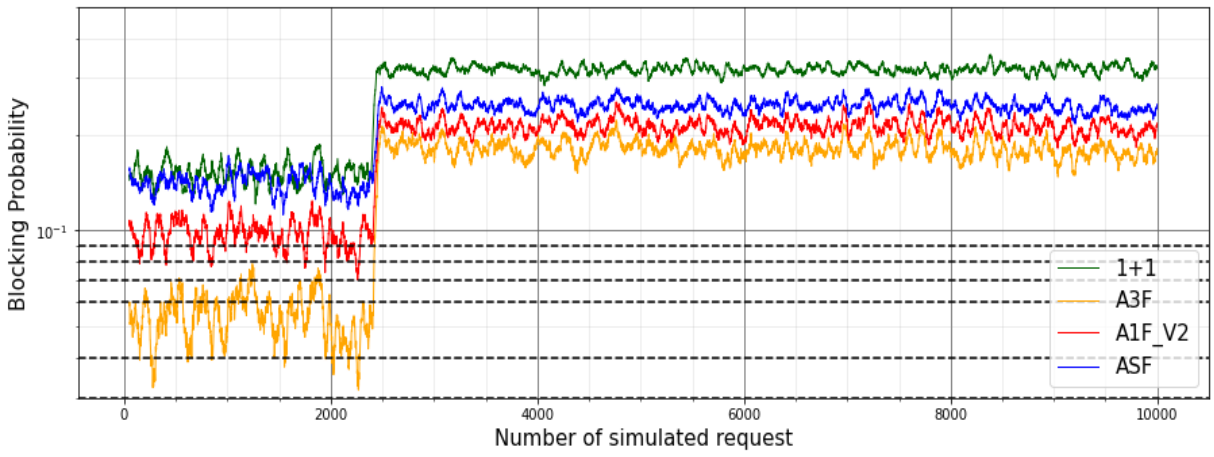

Figura N°18: Evaluación de Agentes y Heurística, con el A1F Mejorado.

Agente / Heurística	Número de Usuarios	[%]
Estado Normal	500.000	100
ASF	399.815	79,96
A1F	410.455	82,09
A1F_V2	418.000	83.60
A3F	433648	86,73

Tabla 8:Comparación de Conexiones Aceptadas entre A1F y A1F_V2.

Visualizando la Figura N°18, corresponde a la nueva evaluación del agente A2C, pero entrenado de tal manera de no presentar olvido catastrófico. En este caso, comparado con la Figura N°16, se aprecia que el agente A1F_V2 presenta una menor probabilidad de bloqueo con respecto al agente A1F. Esta disminución corresponde a un 14,08 %, lo anterior implica un aumento en la capacidad la cual se puede ver en la Tabla 8. Este agente A1F_V2 entrenado aumenta la cantidad de conexiones realizadas en un 1,6% lo cual corresponde a 8.000 nuevas conexiones. Esta diferencia es importante considerando el caso anterior el A1F fue entrenado 25 veces y este agente fue entrenado solo 5 lo cual disminuye los costos computacionales, además disminuyendo los tiempos de iteración, ya que cada iteración de entrenamiento tiene una duración aproximada de 2 hrs. con 40 min. Todo lo comentado anteriormente, valida el hecho que entrenar un agente en un entorno con falla obtiene significativas diferencias con respecto al mismo agente entrenado bajo un entorno de operación sin fallas, siempre y cuando no se produzca olvido catastrófico.

6.3. Política de Agente V/S Heurística.

Dado que la evaluación de los agentes presento buenos resultados, uno de los últimos puntos a comentar sería indicar como estos agentes actúan según el espacio de acción que se les presentaba y compararlo con el algoritmo base con el que fue entrenado el agente.

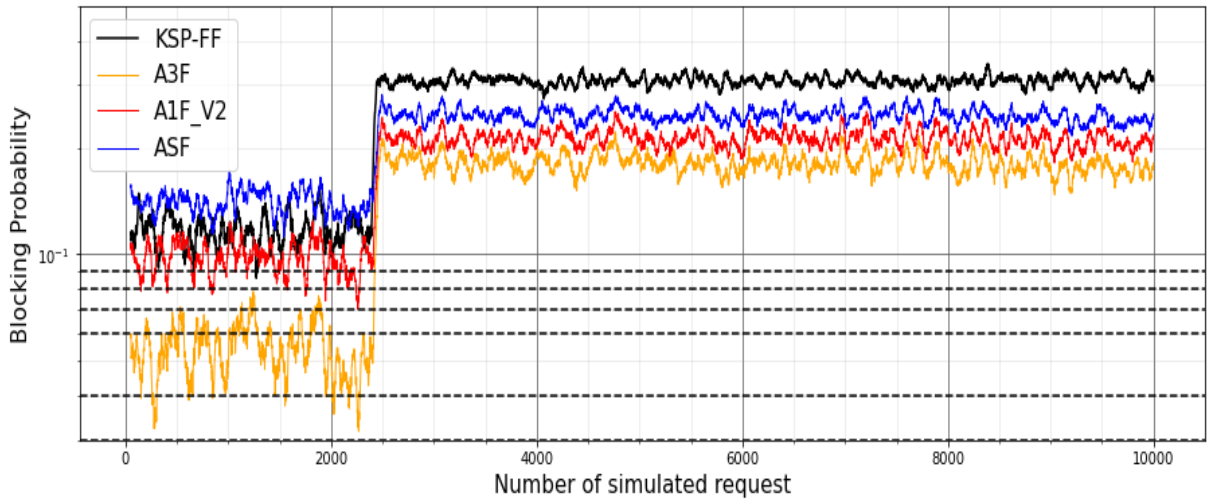

Figura N°19: Evaluación de algoritmo base de entrenamiento de los agentes v/s evaluación de los agentes con el agente A1F_V2.

En la Figura N°19, se visualiza la comparación de comportamiento del algoritmo base de entrenamiento del agente con el mismo entorno que fue presentado en 6.2, en donde la heurística de KSP-FF con restauración presenta una menor probabilidad de bloqueo que en el caso de ASF, pero una peor BP que los agentes A1F y A3F en el escenario previo a la falla. Esto nos indica que el agente ASF no fue capaz de presentar una mejor explotación del aprendizaje adquirido al momento del entrenamiento, aun cuando el espacio de acción del agente ASF es el mismo que el de la heurística KSP-FF, por lo que este agente aplico un proceso de exploración el cual puede ser mejorado, como es el caso de los agentes A1F y A3F, para esto se deberá visualizarlas acciones tomadas por estos agentes, lo anteriormente, es común en casos de entrenamiento de agentes DRL[51].

Por otra parte, en el caso de la falla la heurística KSP-FF, presenta la peor probabilidad de bloqueo que el resto de los agentes. Lo anterior, se debe a que el algoritmo de KSP-FF aunque se pueda aplicar a los problemas de restauración en las fallas. Esta heurística está enfocada en la asignación de recursos espectrales, por lo que previo a la falla esta heurística presenta un mejor comportamiento, y post falla este algoritmo empeora significativamente, ya que sus principales opciones correspondientes a las rutas más cortas se ven alteradas por la generación de la falla.

Dado que todos los agentes presentan una mejor probabilidad de bloqueo, los agentes tienen las mismas ventajas comparativas que la heurística 1+1 expuesta en la sección anterior.

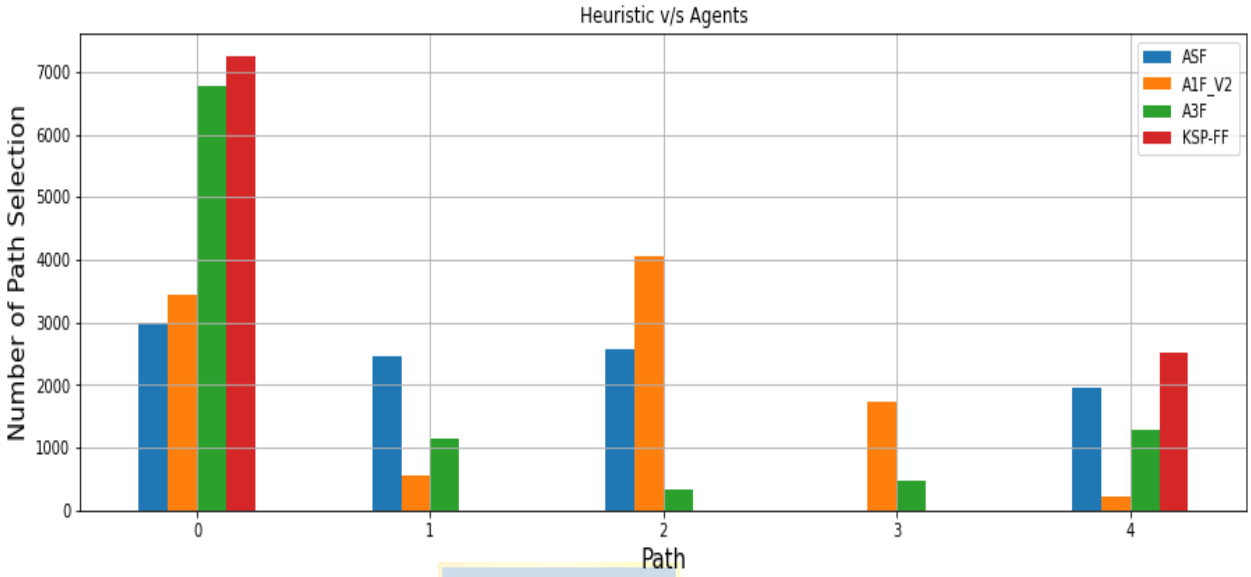

Figura N°20: Comparación de Caminos seleccionados de la Heurística y Agentes.

La Figura N°20, nos indica los caminos seleccionados por el algoritmo base con el que fue entrenado los agentes en distintos entornos. Lo anterior, nos da una idea de cómo estos agentes actúan frente a la misma situación, pero bajo entornos de entrenamiento distintos. Para el primer caso, dado que el agente ASF fue entrenado en un entorno dinámico de operación normal, presenta un comportamiento con notoria distribución equitativa en los caminos seleccionados, demostrando la gran capacidad exploración que tienen los algoritmos de RL. Por otra parte, en el caso de A1F, la distribución de los caminos seleccionados ya no presenta una distribución equitativa de los caminos, haciendo una principal selección entre los caminos 0 y 2 de la figura, esto nos indica que el proceso de explotación está siendo aplicado con éxito según las diferentes situaciones que está enfrentando este A1F. Finalmente, el agente A3F presenta una distribución acentuada entre los caminos 0 y 4, A3F presenta la mejor explotación del conocimiento adquirido por el agente en el entrenamiento ya que, realiza una exploración para todo su espacio de acción. Este procedimiento, se vio fortalecido dado que este agente fue el que se le presento la mayor cantidad de fallas optando por tomar los caminos más cortos , además su comportamiento de asignación de los caminos es el más parecido al de la heurística KSP-FF con la que fue entrenada, validando la idea que si entrenamos agentes en entornos con

fallas, el agente presentará un mejor desempeño que su algoritmo base para el entrenamiento, ya que este agente presenta un proceso de exploración que ayuda a tomar la mejor decisión dada por la mayor probabilidad de acción.

Cabe destacar, que las rutas seleccionadas y mostradas en la Figura N°20 siguen el mismo esquema de selección de rutas expuesto en la sección 5.3.

7. Sumario y Conclusiones

7.1. Sumario

En el presente trabajo se implementó un modelo de aprendizaje reforzado profundo para la resolución del problema de sobrevivencia, enrutamiento y asignación de recursos de la EON (S-RMLSA) NSFNET. Para esto, se expandieron las capacidades del framework Optical RL-Gym diseñada por Natalino, el cual nos daba la capacidad de simular la operación de una red óptica, a este framework se le agrego la capacidad de generar fallas, visualizarlas y obtener las métricas necesarias para su evaluación. Para el desarrollo de esta idea se tomó un agente A2C el cual fue entrenado en 3 entornos distintos, el primer entorno correspondió a uno sin fallas, por lo que este agente fue entrenado en un entorno de operación normal. El segundo entorno correspondió a una operación dinámica, pero con una falla, este proceso de falla se repitió por cada enlace que estaba presente en la red óptica más una última operación en operación dinámica normal. Finalmente, el ultimo agente fue entrenado en un entorno en el cual se presentan 3 fallas en distintos tiempos y en donde cada falla corresponde a enlaces distintos, en donde uno de los enlaces corresponderá a un enlace central y los otros dos corresponderán a enlaces de borde.

7.2. Conclusiones

El principal desafío de este proyecto radicó en la implementación de un modelo de DRL para la resolución del problema de sobrevivencia, enrutamiento, modulación y asignación de recursos (S-RMLSA) de la EON.

Los resultados obtenidos revelaron que los agentes entrenados en entornos con fallas presentan una mejor distribución de los recursos espectrales superando a las heurísticas de comparación 1+1 con esquema de protecciones dedicadas. El agente ASF presento una mejora promedio del 14.99% con respecto a la BP, mientras que en el caso del agente A1F presentó una mejora promedio del 26.23 %. Por otro lado, el agente A3F fue el que presento la mejor optimización de recursos de la red, con una mejora promedio del 53,99%, lo anterior nos indica

que los agentes tienen una capacidad de adaptación dada por el proceso de exploración que presentan en el entrenamiento, y que, al momento de la evaluación, pueden explotar todo este conocimiento y superar por creces el problema presentado. Lo anterior se comprueba con el aumento de la cantidad de conexiones entregadas por parte de los agente ASF, A1F y A3F del 79,96 % , 82,09% y 86,73% respectivamente ,mientras que la heurística 1+1 solo pudo establecer un 72,82% del total de las conexiones solicitadas.

Por otra parte, la heurística KSP-FF presento resultados relevantes en los cuales obtuvo una menor probabilidad de bloqueo comparado con el agente ASF, lo que nos indica, que el agente bajo un entorno de entrenamiento dinámico y sin falla, no es forzado a explorar todas las alternativas disponibles, por lo que su resultado va a ser parecido o peor que la heurística utilizada como base. En contraste, con respecto al caso de los agentes A1F y A3F presentaron una mejora minimizando la probabilidad de bloqueo la cual optimiza el funcionamiento de la red y aumenta la capacidad de la red mediante la recuperación de conexiones perdidas, en donde las capacidades para el caso de los agentes ASF, A1F y A3F corresponden a 7,14 %, 10,78% y 13,93% respectivamente.

Cabe destacar que, en el proceso de entrenamiento de los agentes, es necesario tener un especial cuidado con respecto al olvido catastrófico observado en el caso del agente A1F, ya que como es usual dentro de las aplicaciones de ML, uno tiende a presentarle a los distintos algoritmos datos en secuencias para sus entrenamientos y estimación de error. Esto en el caso de DRL, puede ser perjudicial, ya que el entorno genera las caracteristicas que aprenderá la red neuronal y el conocimiento que adquiera estará dado tanto por el entorno creado como por la recompensa entregada al agente. Esto cambios se vieron reflejados en la Tabla 8.

Dada las capacidades de los agentes de RL, pueden abordar distintas áreas en el control dinámico de las redes ópticas, por lo que la utilización de un esquema DRL para realizar una tarea concreta dentro de una EON, abre la posibilidad de crear agentes especializados para cada labor. Además, mientras los agentes puedan presentar una mejor visualización del estado global de la red, el algoritmo de DRL superará heurísticas.

El presente documento significó un gran reto personal, ya que su desarrollo requirió la integración de áreas totalmente ajenas a mi carrera de pregrado, y donde las aplicaciones de RL dentro de las redes ópticas están teniendo un constante crecimiento por lo cual debe existir una constante actualización de los conocimientos adquiridos.

Desde abril del 2021, fue necesario aprender y entender ambas áreas (redes óptica-reinforcement learning) para incorporarlas en la implementación de estos algoritmos en la resolución del problema de S-RMLSA, generando una ampliación de la visión sobre el mundo de ciencia de datos y sus próximas aplicaciones en las redes del futuro y como estas tecnologías beneficiaran a la sociedad, impulsando mis habilidades como ingeniero. Además, los objetivos propuestos fueron realizados a lo largo del documento.

Finalmente, la capacidad de información que se estima que tendrá que transportarse por las redes ópticas será fundamental para el desarrollo de la sociedad, entregándole distintos servicios los cuales no podrán ser interrumpidos, por lo que la realización de este estudio y la implementación de nuevas técnicas de ML permitirán redes ópticas más seguras y capaces de soportar fallas de tal manera de poder reestablecer las conexiones y además de entregar las herramientas necesarias para el desarrollo tecnológico y digital de una población. Por lo que esta investigación, podrá tener un alto impacto en un futuro cercano.

7.3. Trabajos a Futuro.

Para futuros trabajos se planea seguir trabajando con técnicas de ML dentro de las comunicaciones ópticas[60] y tomando como base que los agentes entrenados pueden tener buenos resultados en la restauración de fallas. Se planea, utilizar técnicas de las Graph Neural Network, para el monitoreo y manejo de fallas con esquemas de protecciones, de esta manera generar una red inteligente que pueda ser capaz de ajustarse a condiciones de fallas sin necesidad de la intervención de terceros. También otra área que se desea abordar es la utilización de agentes DRL en paralelo para resolver distintas tareas de la red óptica y generar una red inteligente ante distintos casos que se presenten en la EON[61].

8. Bibliografías

[1] S. Gob, “SUBTEL ESTIMA UN AUMENTO DEL 60% EN EL TRÁFICO DE DATOS EN FIESTAS DE FIN DE AÑO POR SALUDOS POR LAS REDES,” *La Nación*, 2020. http://www.lanacion.cl/subtel-estima-un-aumento-del-60-en-el-trafico-de-datos-en-fiestas-de-fin-de-ano-por-saludos-por-las-redes/#:~:text=De acuerdo con las estadísticas,de datos en dicho período.

[2] TeleGeography, “Telegeography,” 2021. https://global-internet-map-2021.telegeography.com/

[3] Ciena, “What it is WDM?,” 2015. https://www.ciena.com.mx/insights/what-is/What-Is-WDM_es_LA.html

[4] H. Kong and C. Phillips, “Improved dynamic lightpath provisioning for large wavelength-division multiplexed backbones,” *Journal of Lightwave Technology*, vol. 25, no. 7, pp. 1693–1701, 2007, doi: 10.1109/JLT.2007.899179.

[5] M. Jinno, “Elastic Optical Networking: Roles and Benefits in beyond 100-Gb/s Era,” *Journal of Lightwave Technology*, vol. 35, no. 5, pp. 1116–1124, 2017, doi: 10.1109/JLT.2016.2642480.

[6] M. Jinno, H. Takara, Y. Sone, K. Yonenaga, and A. Hirano, “Elastic optical path network architecture: Framework for spectrally-efficient and scalable future optical networks,” *IEICE Transactions on Communications*, vol. E95-B, no. 3, pp. 706–713, 2012, doi: 10.1587/transcom.E95.B.706.

[7] H. Waldman, “The Impending Optical Network Capacity Crunch,” *2018 SBFoton International Optics and Photonics Conference, SBFoton IOPC 2018*, pp. 1–4, 2019, doi: 10.1109/SBFoton-IOPC.2018.8610949.

[8] N. Jara, H. Pempelfort, G. Rubino, and R. Vallejos, “A fault-tolerance solution to any set of failure scenarios on dynamic WDM optical networks with wavelength continuity constraints,” *IEEE Access*, vol. 8, pp. 21291–21301, 2020, doi: 10.1109/ACCESS.2020.2967751.

[9] M. To and P. Neusy, “Unavailability Analysis of Long-Haul Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 1, pp. 100–109, 1994, doi: 10.1109/49.265709.

[10] L. Pasteur and R. Koch, “Design of Logical Topologies for Wavelength-Routed Optical Networks,” vol. 74, no. 1934, pp. 535–546, 1941.

[11] R. G. Prinz, A. Autenrieth, and D. A. Schupke, “Dual failure protection in multilayer networks based on overlay or augmented model,” *Proceedings - 2005 DRCN: 5th International Workshop on Design of Reliable Communication Networks - “Reliable Networks for Reliable Services,”* vol. 2005, pp. 179–186, 2005, doi: 10.1109/DRCN.2005.1563863.

[12] O. Gerstel, M. Jinno, A. Lord, and S. J. B. Yoo, “Elastic optical networking: A new dawn for the optical layer?,” *IEEE Communications Magazine*, vol. 50, no. 2, pp. 12–20, 2012, doi: 10.1109/MCOM.2012.6146481.

[13] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka, “Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies,” *IEEE Communications Magazine*, vol. 47, no. 11, pp. 66–73, 2009, doi: 10.1109/MCOM.2009.5307468.

[14] R. Goscien, K. Walkowiak, M. Klinkowski, and J. Rak, “Protection in elastic optical networks,” *IEEE Network*, vol. 29, no. 6, pp. 88–96, 2015, doi: 10.1109/MNET.2015.7340430.

[15] G. Shen, H. Guo, and S. K. Bose, “Survivable elastic optical networks: survey and perspective (invited),” *Photonic Network Communications*, vol. 31, no. 1, pp. 71–87, 2016, doi: 10.1007/s11107-015-0532-0.

[16] Headen Tech, “Comparison of 4 QAM 8 QAM 16 QAM 32 QAM etc.,” 2020. https://www.headendinfo.com/32qam-64qam-128qam-256qam/

[17] NJIT, “16 QAM,” 2020.

[18] F. Shirin Abkenar and A. Ghaffarpour Rahbar, “Study and Analysis of Routing and Spectrum Allocation (RSA) and Routing, Modulation and Spectrum Allocation (RMSA) Algorithms in Elastic Optical Networks (EONs),” *Optical Switching and Networking*, vol. 23, pp. 5–39, 2017, doi: 10.1016/j.osn.2016.08.003.

[19] M. Klinkowski and K. Walkowiak, “Offline RSA algorithms for elastic optical networks with dedicated path protection consideration,” *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, pp. 670–676, 2012, doi: 10.1109/ICUMT.2012.6459751.

[20] X. Chen *et al.*, “Flexible availability-aware differentiated protection in software-defined elastic optical networks,” *Journal of Lightwave Technology*, vol. 33, no. 18, pp. 3872–3882, 2015, doi: 10.1109/JLT.2015.2456152.

[21] M. Troscia, A. Sgambelluri, F. Paolucci, P. Castoldi, P. Pagano, and F. Cugini, “OneM2M IoT Platform for SDN Control of Optical Networks,” no. October, pp. 3–5, 2021.

[22] B. C. Chatterjee, N. Sarma, and E. Oki, “Routing and Spectrum Allocation in Elastic Optical Networks: A Tutorial,” *IEEE Communications Surveys and Tutorials*, vol. 17, no. 3, pp. 1776–1800, 2015, doi: 10.1109/COMST.2015.2431731.

[23] X. Luo, C. Shi, L. Wang, X. Chen, Y. Li, and T. Yang, “Leveraging double-agent-based deep reinforcement learning to global optimization of elastic optical networks with enhanced survivability,” *Optics Express*, vol. 27, no. 6, p. 7896, 2019, doi: 10.1364/oe.27.007896.

[24] J. Mata *et al.*, “Artificial intelligence (AI) methods in optical networks: A comprehensive survey,” *Optical Switching and Networking*, vol. 28, pp. 43–57, 2018, doi: 10.1016/j.osn.2017.12.006.

[25] X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. J. B. Yoo, “DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks,” *arXiv*, vol. 37, no. 16, pp. 4155–4163, 2019.

[26] J. Zhang *et al.*, “A novel shared-path protection algorithm with correlated risk against multiple failures in flexible bandwidth optical networks,” *Optical Fiber Technology*, vol. 18, no. 6, pp. 532–540, 2012, doi: 10.1016/j.yofte.2012.09.002.

[27] H. M. N. S. Oliveira and N. L. S. da Fonseca, “Protection in elastic optical networks using Failure-Independent Path Protecting p-cycles,” *Optical Switching and Networking*, vol. 35, no. May 2019, p. 100535, 2020, doi: 10.1016/j.osn.2019.100535.

[28] H. U. Sheikh and L. Boloni, “Multi-Agent Reinforcement Learning for Problems with Combined Individual and Team Reward,” *Proceedings of the International Joint Conference on Neural Networks*, 2020, doi: 10.1109/IJCNN48605.2020.9206879.

[29] A. M. Hafiz and G. M. Bhat, “Deep Q-Network Based Multi-agent Reinforcement Learning with Binary Action Agents,” *arXiv*, 2020.

[30] M. Guo, J. Lu, and J. Zhou, “Dual-agent deep reinforcement learning for deformable face tracking,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11214 LNCS, pp. 783–799, 2018, doi: 10.1007/978-3-030-01249-6_47.

[31] R. Gu, Z. Yang, and Y. Ji, “Machine learning for intelligent optical networks: A comprehensive survey,” *Journal of Network and Computer Applications*, vol. 157, no. February, p. 102576, 2020, doi: 10.1016/j.jnca.2020.102576.

[32] Y. Zhang, J. Xin, X. Li, and S. Huang, “Overview on routing and resource allocation based machine learning in optical networks,” *Optical Fiber Technology*, vol. 60, no. July, p. 102355, 2020, doi: 10.1016/j.yofte.2020.102355.

[33] Z. Zhao, Y. Zhao, D. Wang, Y. Wang, and J. Zhang, “Reinforcement-learning-based multi-failure restoration in optical transport networks,” *Optics InfoBase Conference Papers*, vol. Part F138-, pp. 5–7, 2019.

[34] Z. Zhao *et al.*, “Service Restoration in Multi-Modal Optical Transport Networks with Reinforcement Learning,” *2020 International Conference on Computing, Networking and Communications, ICNC 2020*, vol. 29, no. 3, pp. 204–208, 2020, doi: 10.1109/ICNC47757.2020.9049800.
[35] P. Morales *et al.*, “Multi-band Environments for Optical Reinforcement Learning Gym for Resource Allocation in Elastic Optical Networks,” *25th International Conference on Optical Network Design and Modelling, ONDM 2021*, 2021, doi: 10.23919/ONDM51796.2021.9492435.
[36] J. Pinto-Ríos *et al.*, “Resource Allocation in Multicore Elastic Optical Networks: A Deep Reinforcement Learning Approach,” *SSRN Electronic Journal*, 2022, doi: 10.2139/ssrn.4075565.
[37] C. Natalino and P. Monti, “The Optical RL-Gym: An open-source toolkit for applying reinforcement learning in optical networks,” pp. 1–5, 2020, doi: 10.1109/icton51198.2020.9203239.
[38] X. Li, T. Gao, L. Zhang, Y. Tang, Y. Zhang, and S. Huang, “Survivable K-Node (Edge) content connected virtual optical network (KC-VON) embedding over elastic optical data center networks,” *IEEE Access*, vol. 6, pp. 38780–38793, 2018, doi: 10.1109/ACCESS.2018.2852814.
[39] A. Anderson *et al.*, “Explaining reinforcement learning to mere mortals: An empirical study,” *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, pp. 1328–1334, 2019, doi: 10.24963/ijcai.2019/184.
[40] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, “Explainability in deep reinforcement learning,” *Knowledge-Based Systems*, vol. 214, p. 106685, 2021, doi: 10.1016/j.knosys.2020.106685.
[41] F. Musumeci, C. Rottondi, G. Corani, S. Shahkarami, F. Cugini, and M. Tornatore, “A Tutorial on Machine Learning for Failure Management in Optical Networks,” *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4125–4139, 2019, doi: 10.1109/JLT.2019.2922586.
[42] M. Liu, M. Tornatore, and B. Mukherjee, “Survivable traffic grooming in elastic optical networks - Shared protection,” *Journal of Lightwave Technology*, vol. 31, no. 6, pp. 903–909, 2013, doi: 10.1109/JLT.2012.2231663.
[43] X. Shao, Y. K. Yeo, Z. Xu, X. Cheng, and L. Zhou, “Shared-path protection in OFDM-based optical networks with elastic bandwidth allocation,” *Optics InfoBase Conference Papers*, pp. 21–23, 2012, doi: 10.1364/ofc.2012.oth4b.4.
[44] M. Klinkowski, “A genetic algorithm for solving RSA problem in elastic optical networks with dedicated path protection,” *Advances in Intelligent Systems and Computing*, vol. 189 AISC, no. May, pp. 167–176, 2013, doi: 10.1007/978-3-642-33018-6_17.
[45] X. Chen, S. Zhu, L. Jiang, and Z. Zhu, “On Spectrum Efficient Failure-Independent Path Protection p-Cycle Design in Elastic Optical Networks,” *Journal of Lightwave Technology*, vol. 33, no. 17, pp. 3719–3729, 2015, doi: 10.1109/JLT.2015.2456052.
[46] M. F. Tuysuz, Z. K. Ankarali, and D. Gözüpek, “A survey on energy efficiency in software defined networks,” *Computer Networks*, vol. 113, pp. 188–204, 2017, doi: 10.1016/j.comnet.2016.12.012.
[47] X. Luo, C. Shi, X. Chen, L. Wang, and T. Yang, “Global optimization of all-optical hybrid-casting in inter-datacenter elastic optical networks,” *IEEE Access*, vol. 6, pp. 36530–36543, 2018, doi: 10.1109/ACCESS.2018.2852067.
[48] R. S. Sutton, A. G. Barto, and A. B. Book, “Reinforcement Learning, 1st ed,” 1998.
[49] C. Paternina-Arboleda, “Reinforcement Learning Scheme,” *Research gate*, 2008. https://www.researchgate.net/publication/228967570_Abstract_FLEXIBLE_STATE-DEPENDANT_MACHINE_SCHEDULING_PROBLEMS_USING_REINFORCEMENT_LEARNING/figures?lo=1)
[50] E. F. Morales and J. H. Zaragoza, “An introduction to reinforcement learning,” *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*, pp. 63–80, 2011, doi: 10.4018/978-1-60960-165-2.ch004.
[51] Y. ALAOUI MRANI, “University College London (UCL) Reinforcement Learning for Survivability in Optical Networks,” no. April, 2021.

[52] M. Tokic, “Adaptive ε-greedy exploration in reinforcement learning based on value differences,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6359 LNAI, pp. 203–210, 2010, doi: 10.1007/978-3-642-16111-7_23.

[53] V. R. Konda and J. N. Tsitsiklis, “Actor-critic algorithms,” *Advances in Neural Information Processing Systems*, pp. 1008–1014, 2000.

[54] Y. Sone *et al.*, “Bandwidth squeezed restoration in spectrum-sliced elastic optical path networks (SLICE),” *Journal of Optical Communications and Networking*, vol. 3, no. 3, pp. 223–233, 2011, doi: 10.1364/JOCN.3.000223.

[55] T. H. Lian Zhang, “Energy-Aware Virtual Machine Management in Inter-Datacenter Networks Over Elastic Optical Infrastructure,” *Research gate*, 2017.

[56] D. A. Bader, S. Kintali, K. Madduri, and M. Mihail, “Approximating betweenness centrality,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4863 LNCS, no. December, pp. 124–137, 2007, doi: 10.1007/978-3-540-77004-6_10.

[57] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, “Stable-baselines3: Reliable reinforcement learning implementations,” *Journal of Machine Learning Research*, vol. 22, pp. 1–8, 2021.

[58] M. Džanko, B. Mikac, and M. Furdek, “Dedicated path protection for optical networks based on function programmable nodes,” *Optical Switching and Networking*, vol. 27, no. September 2017, pp. 79–87, 2018, doi: 10.1016/j.osn.2017.09.001.

[59] J. Kirkpatrick *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proc Natl Acad Sci U S A*, vol. 114, no. 13, pp. 3521–3526, 2017, doi: 10.1073/pnas.1611835114.

[60] W. S. Saif, M. A. Esmail, A. M. Ragheb, T. A. Alshawi, and S. A. Alshebeili, “Machine Learning Techniques for Optical Performance Monitoring and Modulation Format Identification: A Survey,” *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2839–2882, 2020, doi: 10.1109/COMST.2020.3018494.

[61] X. Chen, R. Proietti, C. Y. Liu, and S. J. B. Yoo, “A Multi-Task-Learning-Based Transfer Deep Reinforcement Learning Design for Autonomic Optical Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 9, pp. 2878–2889, 2021, doi: 10.1109/JSAC.2021.3064657.