

SORUS

Validación y optimización conjunta de RIS y vRANs

SORUS-RAN A1.2

CASOS DE USO vRAN

Revisión	Autor	Fecha de entrega	Cambios
Versión 01	Jose Ayala Romero, Andrés García Saavedra	16/05/2023	Versión Inicial
Versión 02	Luis Roda Sánchez, Jorge San Martin Gomez	22/05/2023	revisión y actualización de tablas y resumen ejecutivo y conclusiones.

Exención de responsabilidad:

El apoyo de la Comisión Europea a la elaboración de esta publicación no constituye una aprobación de su contenido, que refleja únicamente las opiniones de los autores, y la Comisión no se hace responsable del uso que se pueda hacer de la información aquí difundida.

CONTENIDOS

LISTA DE ABREVIATURAS Y ACRÓNIMOS.....	3
LISTA DE FIGURAS	5
LISTA DE TABLAS	6
1 RESUMEN EJECUTIVO.....	7
2 INTRODUCCIÓN	8
3 CASOS DE USO	10
3.1. MÉTRICAS	12
3.2. CASOS DE USO 1 Y 2	13
3.2.1. INFORMACIÓN DE CONTEXTO	13
3.2.2. POLÍTICAS DE CONTROL.....	13
3.2.3. FUNCIÓN DE RECOMPENSA	14
3.3. CASO DE USO 3.....	14
3.3.1. INFORMACIÓN DE CONTEXTO	14
3.3.2. POLÍTICAS DE CONTROL.....	14
3.3.3. FUNCIÓN DE RECOMPENSA	15
4 FORMULACIÓN DEL PROBLEMA	16
4.1. CASO DE USO 1.....	16
4.2. CASO DE USO 2.....	17
4.3. CASO DE USO 3.....	17
5 CONCLUSIONES.....	19
6 REFERENCIAS	21

LISTA DE ABREVIATURAS Y ACRÓNIMOS

Tabla 1. Lista de abreviaturas y acrónimos

Abreviatura	Explicación/Definición
AP	Average Precision
AR	Augmented Reality
BBU	Unidad de Banda Base
BS	Base Station
BSR	Buffer State Report
CAPEX	Gastos de Capital
CQI	Channel Quality Indicator
DL	Downlink
GPU	Graphics Processing Unit
IA	Inteligencia Artificial
IoU	Intersección sobre la Unión
LA	Learning Agent
MAC	Media Access Control
mAP	mean Average Precision
MCS	Modulation and Coding Scheme
ML	Machine Learning
MVA	Mobile Video Analytics
OPEX	Gastos Operativos
O-DU	Open – Distributed Unit
PDCP	Packet Data Convergence Protocol
PoE	Power-over-Ethernet
PUCCH	Canal Físico de Control del Enlace Ascendente
PUSCH	Canal Físico Compartido del Enlace Ascendente
RAN	Radio Access Network
RL	Reinforcement Learning

Abreviatura	Explicación/Definición
RIC	RAN Intelligent Controller
RT	Real Time
RU	Unidad de Radio
SMO	Service Management and Orchestration
SNR	Signal-to-Noise Ratio
TPC	Control de Potencia de Transmisión
UCI	Control del Enlace Ascendente
UE	User Equipment
UL	Uplink
vBS	virtualized Base Station
VR	Virtual Reality
vRAN	virtualized Radio Access Network

LISTA DE FIGURAS

FIGURA 1 . ARQUITECTURA DE LOS CASOS DE USO 1 Y 2.....	11
FIGURA 2. ARQUITECTURA DEL CASO DE USO 3.....	12

LISTA DE TABLAS

TABLA 1. LISTA DE ABREVIATURAS Y ACRÓNIMOS.....3

1 RESUMEN EJECUTIVO

Este documento corresponde al entregable A1.2 – Casos de uso vRAN – que se encuentra en la fase inicial de las tareas descritas en el pliego de condiciones técnicas. Se describen tres casos de uso de vRAN relevantes y se estudian los parámetros que interesa optimizar, las métricas a tener en cuenta y los distintos objetivos de optimización. Tras la introducción, el objetivo principal es identificar los escenarios en los que la virtualización vRAN sea una ventaja respecto a los despliegues existentes. Se estudia el despliegue de múltiples O-DU virtualizadas sobre los nodos de computación compartida, así como las ventajas, desventajas y retos de desplegar diferentes O-DUs virtualizadas en la misma plataforma. También se describen los retos técnicos que presenta desplegar múltiples O-DUs a nivel computacional y aceleración de hardware compartidos, así como servicios de IA asociados en el Edge. Se analiza el uso tanto de controladores Non-Real Time como Near-Real Time para mejorar la asignación de recursos según la demanda.

2 INTRODUCCIÓN

La implantación de redes móviles 5G, así como el futuro de las mismas, reportará grandes beneficios a los usuarios, pero, a su vez, con el objetivo de aprovechar todo su potencial, también implica la implementación de ciertos avances técnicos inexistentes hasta este momento. La arquitectura y componentes introducidos permiten una flexibilidad, tiempos de respuesta y ancho de banda sin precedentes. Un elemento fundamental para conseguirlo, dentro de esta nueva arquitectura, es la computación en el borde, o Edge computing. Gracias a la utilización de pequeños servidores, lo más cerca posible de las fuentes de datos, se consigue reducir el tráfico hacia y desde la red troncal y la latencia en las comunicaciones.

Sin embargo, esto es sólo el principio si pretendemos sacar el máximo partido a los recursos actuales y adaptarnos al contexto y los requerimientos de los usuarios en cada momento. La virtualización se considera uno de los enfoques clave para que las redes celulares estén a la altura de los exigentes servicios que aspiran a ofrecer a los usuarios [1]. La última frontera en este empeño es el desarrollo de redes de acceso radioeléctrico virtualizadas (vRAN), en las que las estaciones base (BS) tradicionales se sustituyen por pilas virtualizadas, como las desarrolladas por srsRAN [1] y OpenAirInterface (OAI) [3]. Estas novedosas BSs virtualizadas (vBS) son totalmente configurables y pueden desplegarse en distintas plataformas, desde servidores básicos y pequeños dispositivos integrados hasta nodos móviles como drones [4]. Esta transformación de la RAN constituye un cambio de paradigma para las redes celulares y se espera que ofrezca la tan necesaria flexibilidad de rendimiento, facilite la necesaria densificación de la red y reduzca significativamente sus gastos de CAPEX Y OPEX [5]. Por lo tanto, no es de extrañar que hoy en día veamos numerosos esfuerzos de la industria con el objetivo de construir tales pilas de software [1], diseñar arquitecturas RAN totalmente abiertas [6], e incluso llevar a cabo extensas pruebas de campo [7].

Sin embargo, la llegada de las vRAN plantea nuevos retos técnicos, ya que las estaciones base virtualizadas (vBS) difieren significativamente de las estaciones base basadas en hardware. Por un lado, las soluciones Open RAN (lideradas por la alianza O-RAN) permiten a las vBS cambiar en tiempo real diversos parámetros de funcionamiento, como la potencia de transmisión y los esquemas de modulación, para adaptarse a las volátiles condiciones de la red y a las dinámicas necesidades de los usuarios.

Por otro lado, aunque esto proporciona a los operadores de red un nivel de flexibilidad sin precedentes, tiene el coste de un rendimiento menos predecible debido a los complejos acoplamientos entre el espacio de posibles configuraciones (cuya dimensionalidad es muy alta), pudiendo afectar al rendimiento resultante.

Esto último es crucial por razones económicas, especialmente a la luz de la creciente densificación de la red; pero también porque a menudo se espera que las vBS operen con potencia limitada. Pensemos, por ejemplo, en las vBS que funcionan con baterías o con líneas Power-over-Ethernet (PoE). Por lo tanto, las políticas de control de recursos existentes corren el riesgo de infrautilizar este nuevo tipo de BS, o de hacer que las vRAN sean económicamente insostenibles.

Por otro lado, existe un consenso cada vez mayor sobre la necesidad de que la próxima generación de redes móviles admita inteligencia artificial (IA) y otros servicios inteligentes en el borde de la red (network Edge). Estos servicios suelen requerir la recopilación, transferencia y procesamiento de flujos de datos en tiempo real, con el objetivo de proporcionar operaciones de datos, por ejemplo, inferencias, a usuarios finales como pequeños dispositivos IoT, drones o smartphones sobre la

marcha. Un ejemplo representativo de estos servicios son los análisis de vídeo móvil o *mobile video analytics* (MVA), que se utilizan en servicios de realidad aumentada o realidad virtual AR/VR [10] aplicaciones de asistencia cognitiva [11], sistemas de vigilancia [12], entre otros servicios de IA similares. La tarea central de MVA es que los dispositivos de usuario envíen fotogramas de vídeo a la red, que debe procesarlos y transmitir de vuelta objetos representados detectados con precisión, o extraer otra información importante [13].

Aunque los servicios de tipo MVA ya se consideran una utilidad de la que cada usuario debería poder disfrutar, su amplia implantación exige un cambio fundamental en la forma en que gestionamos las redes móviles. En estos servicios, el papel de la red no se limita a transferir datos de un punto a otro, ni siquiera a procesarlos en ruta. En su lugar, la red tiene que optimizar directamente el rendimiento del servicio, lo que implica los criterios de precisión (inferencias sin errores), latencia de extremo a extremo (inferencias rápidas) y rendimiento de la tarea (inferencias/s) de una manera eficiente en cuanto a recursos. Este último requisito es crucial, ya que estos servicios crean flujos de datos voluminosos, implican cálculos pesados y consumen grandes cantidades de energía. De hecho, el consumo de energía no solo es uno de los gastos de explotación más frecuentes de las redes móviles¹ [14], pero también se ha identificado como el principal obstáculo para el éxito de estos servicios inherentemente demandantes de energía. Además, la energía es el recurso común consumido por todas las operaciones de red (por ejemplo, transmisión, transferencia o computación de datos) y su gestión eficiente es imperativa también desde el punto de vista del rendimiento.

Está claro que para desplegar estos servicios necesitamos desarrollar una metodología sistemática de control energético de los recursos de comunicación y computación de la red. Para ello, es imprescindible comprender primero cómo funcionan estos servicios, qué parámetros del sistema determinan su rendimiento y sus necesidades de recursos, si hay puntos dulces en la función rendimiento/coste y cómo optimizar la red y el servicio para aprovechar estas oportunidades.

En concreto, nos enfrentamos a varios retos a la hora de abordar este problema. Por ejemplo, no existe una configuración óptima única para el sistema. La configuración óptima depende del contexto (condiciones del canal inalámbrico, número de usuarios, etc.) y cambia con el tiempo. Por ejemplo, unos enlaces inalámbricos más deficientes pueden aumentar el retardo (debido a la necesidad de niveles de modulación más bajos que hacen disminuir la velocidad de transmisión de datos) y el sistema tiene que compensarlo asignando más recursos radioeléctricos (ancho de banda o tiempo de canal), lo que aumenta los requerimientos computacionales de la estación base. Otra opción para compensar el retraso es aumentar la capacidad computacional en el servidor Edge, de modo que se reduzca el tiempo de procesamiento. En consecuencia, tenemos que encontrar la correspondencia entre los contextos y las configuraciones óptimas. Esto no es sencillo porque la relación entre las configuraciones y los indicadores de rendimiento es no lineal y desconocida a priori.

Además, el sistema considerado debe gestionar un número variable de usuarios. Una variación en el número de usuarios suele implicar un cambio en la dimensionalidad del problema, lo que supone un gran reto para los modelos de Machine Learning (ML). A su vez, un aumento de la

¹ Por ejemplo, China Mobile se comprometió a reducir el consumo total de energía por unidad de negocio de telecomunicaciones en no menos del 6% en 2021 [15], y Verizon y Vodafone han fijado objetivos para alcanzar las emisiones netas de energía cero en 2040 [16].

dimensionalidad del problema puede reducir el tiempo de convergencia debido a la maldición de la dimensionalidad² (*curse of dimensionality*). Por tanto, es necesario proponer una estrategia novedosa para gestionar el número de usuarios.

3 CASOS DE USO

Consideramos diferentes casos de uso de vRAN que son relevantes para el diseño de redes móviles de próxima generación:

- Caso de uso 1:** equilibrio entre rendimiento y coste. Consideramos una vBS compuesta por una unidad de banda base (BBU) que puede corresponder a un eNodeB (eNB) 4G o un gNodeB (gNB) 5G alojado en una plataforma de computación y conectado a una unidad de radio (RU). Este tipo de BS es relevante para células pequeñas de bajo coste, células PoE y otras plataformas similares que son cada vez más comunes en las redes 5G y posteriores. La Figura 1 muestra la arquitectura del sistema, que es compatible con las especificaciones de O-RAN [17]. El agente de aprendizaje o Learning Agent (LA) implementa algoritmos de aprendizaje automático dentro del RAN Intelligent Controller (RIC) con granularidad de segundos (Non-RT RIC) en el orquestador del sistema, y selecciona políticas de control radio que son eficientes para cada periodo de orquestación $t = 1, \dots, T$ (normalmente del orden de segundos). La decisión óptima (es decir, una política de radio) en cada t depende de la información de contexto. Esta es proporcionada al principio de cada periodo por el vBS (a través de la interfaz O1) a partir de las mediciones recopiladas con una granularidad por debajo de los segundos dentro del Near-RT RIC (utilizando la interfaz E2). Las políticas de control radio computadas son entonces configuradas en el vBS a través de su interfaz A1-P como se muestra en la Figura 1. Al final de cada periodo de orquestación, el módulo Data Monitor en el Near-RT RIC calcula un valor de recompensa agregando las métricas de rendimiento adoptadas, que se recogen del vBS a través de la interfaz E2; y finalmente proporciona los resultados al LA (interfaz O1).
- Caso de uso 2:** limitación en la máxima potencia que la estación base virtualizada puede usar. En este caso de uso consideramos que el vBS se alimenta de una fuente de energía

² [HTTPS://LINK.SPRINGER.COM/REFERENCEWORKENTRY/10.1007/978-0-387-39940-9_133](https://link.springer.com/referenceworkentry/10.1007/978-0-387-39940-9_133)

limitada en potencia como puede ser una batería o una fuente PoE. La arquitectura también se representa en la Figura 1 **Error! Reference source not found..**

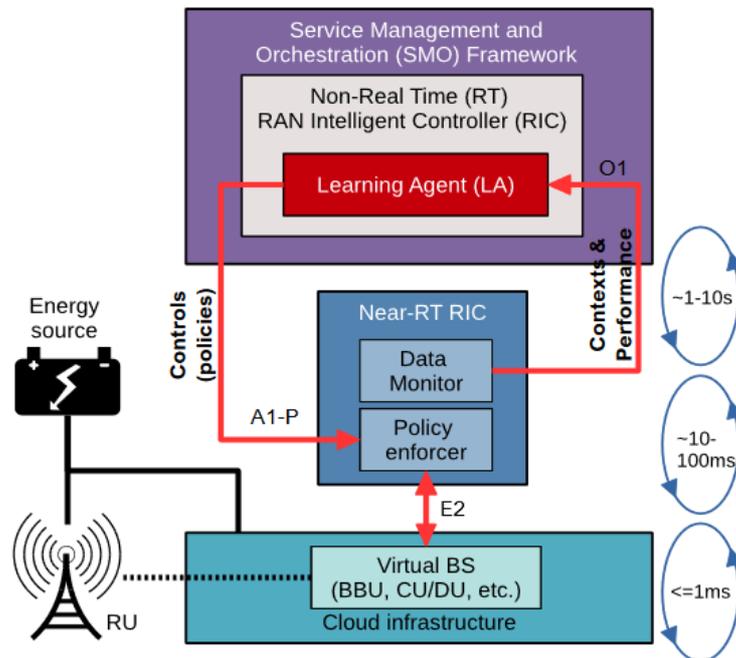


Figura 1 . Arquitectura de los casos de uso 1 y 2

- **Caso de uso 3:** optimización conjunta de vBS y servicios Edge de IA. Consideramos un servidor al borde de la red (*network Edge*) que dispone de una GPU que proporciona un servicio de IA a través de una red de acceso por radio. El servicio de IA puede ser, por ejemplo, un servicio de reconocimiento de objetos que puede utilizarse para la vigilancia de la seguridad o la detección de fallos en cadenas industriales. Suponemos que se crea un slice dedicado a este servicio que incluye la vBS y el servidor de borde de la red [8]–[9]. Esto se ilustra con los recuadros naranjas de la Figura 2. El funcionamiento del servicio es el siguiente: los usuarios capturan imágenes que se envían al servidor de borde a través del enlace ascendente de la interfaz de radio del vBS. A continuación, la GPU del servidor procesa los datos entrantes y genera una respuesta, que se devuelve a los usuarios a través del enlace descendente del vBS.

El flujo de datos es el siguiente: el LA observa periódicamente el contexto (definido más adelante), orquesta los recursos asignados al acceso inalámbrico y al servicio GPU mediante un conjunto de políticas de control, y utiliza una función de recompensa que agrega indicadores clave de rendimiento del sistema para tomar mejores decisiones a lo largo del tiempo. Para ello, seguimos el esquema de O-RAN [17], una alianza de operadores y

fabricantes liderada por los operadores para construir soluciones RAN abiertas e inteligentes [18].

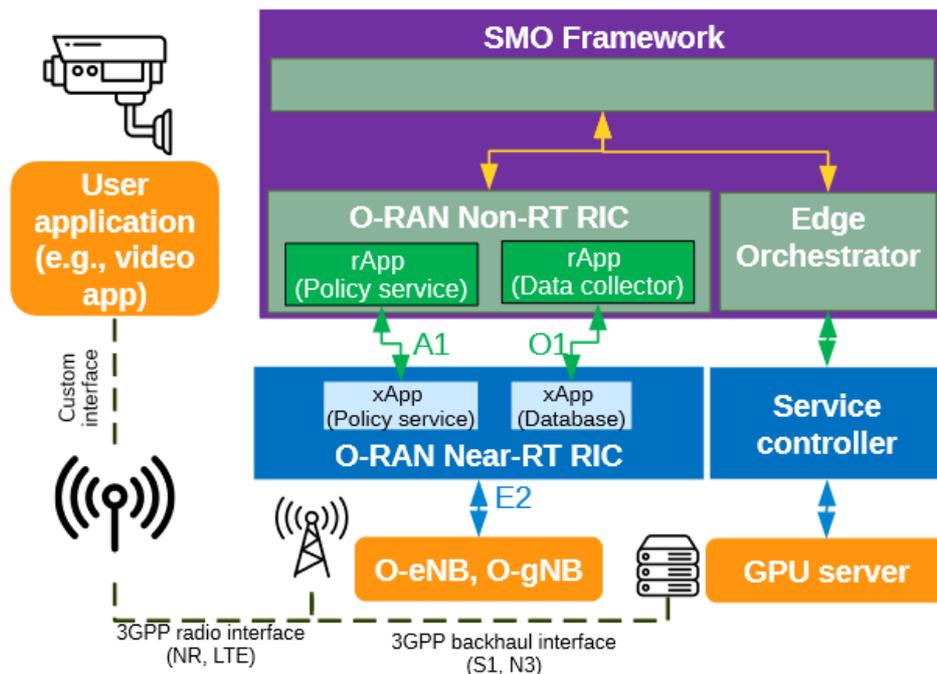


Figura 2. Arquitectura del caso de uso 3

Nuestro objetivo final en todos los casos de uso es utilizar la arquitectura de control de O-RAN para implementar políticas de configuración que se adapten a la dinámica del sistema y, al mismo tiempo, satisfagan diferentes criterios de rendimiento, prestando siempre especial atención al consumo de energía. Las políticas de configuración son seleccionadas por un agente de aprendizaje o LA desplegado en el marco SMO, como se muestra en la Figura 1.

3.1. Métricas

En esta sección definimos las diferentes métricas utilizadas para caracterizar los diferentes casos de uso y formular los problemas de aprendizaje automático. Definimos diferentes tipos de métricas:

- **Información de contexto:** se refiere a métricas usadas como entrada a los algoritmos de aprendizaje automático para la toma de decisiones y, usualmente, caracterizan el estado de la red.
- **Políticas de control:** hace referencia a los parámetros que deben ajustarse por los algoritmos de aprendizaje automático para optimizar la red de acuerdo con los objetivos que se definan.
- **Funciones de recompensa:** hacen referencia a métricas de rendimiento o a funciones de una o varias de éstas. Son usadas para entrenar los algoritmos de aprendizaje automático reforzando comportamientos que estén alineados con estas funciones de recompensa.

En la siguiente sección se definen las diferentes métricas para cada uno de los casos de uso.

3.2. Casos de Uso 1 y 2

3.2.1. Información de contexto

Definimos el contexto del canal de bajada o downlink (DL) en cada período t como $\omega_t^{dl} := [\overline{c_t^{dl}}, \widetilde{c_t^{dl}}, d_t^{dl}]$, donde $\overline{c_t^{dl}}$ y $\widetilde{c_t^{dl}}$ son la media y la varianza del indicador de calidad del canal DL (CQI o Channel Quality Indicator) de todos los usuarios en el periodo anterior; y d_t^{dl} son los nuevos bits que llegan a través del DL de la vBS de todos los usuarios agregados. Obsérvese que los valores de CQI de DL se envían periódicamente desde los equipos de usuario (UE) a vBS a través de la información de control del enlace ascendente (UCI) transportada por el canal físico compartido del enlace ascendente (PUSCH) o el canal físico de control del enlace ascendente (PUCCH) de 4G/5G. A la inversa, d_t^{dl} se mide mediante la vBS en la capa de protocolo de convergencia de paquetes de datos (PDCP).

Asimismo, definimos el contexto del canal de subida o uplink (UL) como $\omega_t^{ul} := [\overline{c_t^{ul}}, \widetilde{c_t^{ul}}, d_t^{ul}]$. El CQI del UL se mide mediante el vBS en la capa MAC, y las nuevas llegadas de bits al UL se estiman a partir de los informes periódicos de estado del búfer (buffer state report o BSR) de los UE. Todas estas medidas son recogidas por el monitor de datos del RIC Near-RT (ver Figura 1) desde el vBS utilizando la interfaz E2 con una granularidad por debajo de los segundos, y son agregadas al comienzo de cada periodo de orquestación t . Denotamos el vector de contexto global $\omega_t := [\omega_t^{dl}, \omega_t^{ul}] \in \Omega$, donde Ω es el espacio de contexto.

Obsérvese que los contextos están relacionados con la carga de tráfico y la calidad del canal y son parámetros exógenos, es decir, las decisiones de configuración (definidas a continuación) no les afectan.

3.2.2. Políticas de control

Definimos las políticas de control del canal de bajada o downlink (DL) como $x_t^{dl} := [p_t^{dl}, m_t^{dl}, a_t^{dl}]$ en el periodo t , donde $p_t^{dl} \in \mathcal{P}^{dl}$ es una política de control de potencia de transmisión (TPC) para la máxima potencia de transmisión vBS permitida, $m_t^{dl} \in \mathcal{M}^{dl}$ es el Modulation and Coding Scheme (MCS) más alto admisible por la vBS (*política de MCS de DL*), y $a_t^{dl} \in \mathcal{A}^{dl}$ es el tiempo máximo de transmisión de vBS (*política de tiempo de transmisión DL o airtime de DL*).

Definimos las políticas de control del canal de subida o uplink (UL) como $x_t^{ul} := [m_t^{ul}, a_t^{ul}]$, donde $m_t^{ul} \in \mathcal{M}^{ul}$ y $a_t^{ul} \in \mathcal{A}^{ul}$ son las políticas UL MCS y de airtime³. Por tanto, formalizamos cada control en el periodo de decisión t como una *política de radio*:

$$x_t := [x_t^{dl}, x_t^{ul}] \in \mathcal{X}, \quad \mathcal{X} = \mathcal{P}^{dl} \times \mathcal{M}^{dl} \times \mathcal{A}^{dl} \times \mathcal{M}^{ul} \times \mathcal{A}^{ul},$$

donde \mathcal{X} es el espacio de control. Una vez calculada, el LA envía cada política de control de radio al RIC Near-RT a través de la interfaz A1-P de la O-RAN, que luego se aplica al vBS. Las políticas UL se aplican configurando cada programación UL en la capa MAC de vBS.

³ No definimos una política UL TPC ya que la potencia de transmisión de los usuarios tiene menos impacto en la potencia vBS que el MCS y el tiempo aire UL; pero nuestro marco puede ampliarse fácilmente para incluir esta decisión.

3.2.3. Función de recompensa

Denotamos con $R^{dl}(\omega_t^{dl}, x_t^{dl})$ y $R^{ul}(\omega_t^{ul}, x_t^{ul})$ las velocidades de transmisión de datos (throughput) de DL y UL, y definimos la función de recompensa:

$$r(\omega_t, x_t) := \log\left(1 + \frac{R^{dl}(\omega_t^{dl}, x_t^{dl})}{d_t^{dl}}\right) + \log\left(1 + \frac{R^{ul}(\omega_t^{ul}, x_t^{ul})}{d_t^{ul}}\right)$$

Donde los logaritmos se utilizan para lograr la equidad entre los flujos DL y UL – y para ello, se podría utilizar cualquier otra función α -fair [19]. Obsérvese que dividimos las tasas alcanzadas con la carga real en el flujo respectivo (enlace ascendente o descendente), ya que la recompensa debe definirse naturalmente en relación con las necesidades del sistema. Además, es importante destacar que, en la práctica, sólo podemos esperar observar *valores ruidosos* de estas funciones, incluso cuando sus argumentos son fijos, porque, naturalmente, el funcionamiento del sistema es estocástico y también las mediciones de potencia son ruidosas, como veremos en el entregable SORUS-RAN-A2.1. Afortunadamente, nuestro marco de optimización puede hacer frente a estos problemas. En adelante, denotaremos con $R_t^{dl}(\omega_t^{dl}, x_t^{dl})$, $R_t^{ul}(\omega_t^{ul}, x_t^{ul})$ y $r_t(\omega_t, x_t)$ estas muestras ruidosas de las funciones en el periodo t , que se consideran estacionarias y devuelven los valores medios (no perturbados) respectivos cuando se promedian (es decir, en esperanza).

3.3. Caso de uso 3

3.3.1. Información de contexto

Definimos el contexto en cada periodo de tiempo t como $c_t := [n_t, \bar{c}_t, \tilde{c}_t] \in \mathcal{C}$, donde n_t es el número de usuarios del slice, y \bar{c}_t y \tilde{c}_t son la media y la varianza del UL CQI de todos los usuarios del segmento durante el periodo anterior, y \mathcal{C} es el espacio de contexto. Nótese que consideramos una caracterización estadística de los estados de canal de los usuarios, lo que evita problemas de dimensionalidad en el diseño de los algoritmos de aprendizaje automático.

3.3.2. Políticas de control

Resolución de las imágenes: esta política de control establece la codificación media de cada imagen (número de píxeles) que puede aplicar el servicio. En nuestros experimentos, la resolución máxima (100%) es de 640x480 píxeles. Nótese que, en un momento dado, la resolución de una imagen puede ser mayor o menor que la seleccionada por la política de control, siempre que la media de todo el periodo y de los usuarios respete este valor.

Tiempo de transmisión o airtime: esta política de radio impone una restricción a los recursos de radio (ciclo de trabajo) que el vBS asigna al tráfico de servicio. El scheduler de radio de la capa MAC, que funciona con una granularidad de milisegundos, debe asignar los recursos de radio (que pueden variar según los usuarios en función de sus canales) de forma que se respete el valor establecido por esta política de control. Debido a la naturaleza de este servicio, nos centramos en la comunicación de enlace ascendente (uplink).

Velocidad de la GPU: esta política de control establece un límite de potencia de la GPU que adapta la velocidad de procesamiento de una GPU (o un grupo de GPUs) en un slice para cumplir con la restricción de potencia adoptada. El controlador de la GPU (por ejemplo, el controlador de NVIDIA) puede cambiar la velocidad de la GPU en cualquier momento (por ejemplo, para diferentes fotogramas de vídeo) siempre que se respete la velocidad media establecida por esta política.

Modulation and Coding Scheme: esta política impone una restricción al MCS máximo elegible por la vBS para transportar los datos del servicio por el aire. El MCS seleccionado por la capa MAC puede ser inferior a este límite para algunos usuarios en función de su estado de canal (SNR/CQI).

Definimos \mathcal{H} como el conjunto de posibles resoluciones de imagen; \mathcal{A} el conjunto de posibles configuraciones de airtime (recursos de radio de enlace ascendente) que pueden asignarse; Γ las posibles configuraciones de velocidad de GPU; y \mathcal{M} el conjunto de todas las posibles políticas MCS (que caracterizan las velocidades de datos) definidas anteriormente. Por lo tanto,

$$x_t := [\eta_t, a_t, \gamma_t, m_t] \in \mathcal{X} := \mathcal{H} \times \mathcal{A} \times \Gamma \times \mathcal{M}$$

denota la política de control seleccionada en el periodo de tiempo t . La velocidad de la GPU se configura en la misma máquina donde se ejecuta el agente de aprendizaje, el airtime y las políticas MCS pueden enviarse al vBS a través de la interfaz A1-P de la arquitectura O-RAN, y la resolución de la imagen se indica al usuario mediante la aplicación del servicio.

Suponemos que la tecnología RAN puede configurarse en términos de dominio temporal y velocidad de datos (airtime y MCS en esta formulación, respectivamente). Otras tecnologías de radio quedan fuera del ámbito de este documento.

Nos centramos en las políticas radioeléctricas de enlace ascendente (uplink) porque, como confirman nuestros experimentos desarrollados para SORUS-RAN-A2.1, estos servicios de IA tienen escaso impacto en el enlace descendente, ya que la mayoría de datos suele ir en sentido ascendente (uplink) y sólo fluye información simple (por ejemplo, etiquetas de las imágenes) en sentido descendente (downlink).

Nótese que las políticas en \mathcal{X} controlan conjuntamente parámetros del dispositivo de usuario, el vBS y el servidor Edge. Estos tres elementos están altamente acoplados (como mostramos en el documento de SORUS-RAN-A2.1) y por esa razón deben configurarse a la vez.

3.3.3. Función de recompensa

Tiempo de servicio: retraso de extremo a extremo que incluye el preprocesamiento de la imagen en el lado del usuario, su transmisión, el procesamiento en el servidor (retraso de la GPU) y la devolución de los cuadros delimitadores y las etiquetas.

Precisión media (mAP): la precisión del servicio se cuantifica mediante la Precisión Media o Mean Average Precision (mAP) [20]. Por un lado, la precisión se define como la proporción de verdaderos positivos sobre todas las clasificaciones positivas. Por otro lado, el recall mide lo bien que se identifican estos positivos calculando el ratio entre verdaderos positivos sobre la suma de verdaderos positivos y falsos negativos. La intersección sobre la unión (IoU) mide el solapamiento entre el cuadro delimitador calculado y las etiquetas. Los valores de IoU por encima de un umbral (fijado aquí en 0,5) desencadenan un verdadero positivo. A continuación, para un conjunto determinado de imágenes, la precisión media (Average Precision, AP) corresponde al área por debajo de la curva precisión-recall. Por último, mAP es la media de AP de todas las categorías de objetos, por lo que oscila entre 0 (peor rendimiento) y 1 (mejor rendimiento).

Consumo energético del servidor: coste energético asociado a la carga computacional de las peticiones del servicio, que está dominado por el consumo energético de la GPU.

Consumo energético de la vBS: consumo de energía asociado al procesamiento de la unidad de banda base (BBU) en un entorno RAN virtualizado.

El tiempo del servicio experimentado por el usuario i se denota por $D_i(c, x)$, y el mAP se denota por $Q_i(c, x)$. A continuación, definimos:

$$d(c, x) := \max_i D_i(c, x), \quad \text{and} \quad \rho(c, x) := \min_i Q_i(c, x),$$

siendo el mayor retardo y el menor mAP, respectivamente, entre todos los usuarios. La potencia consumida en el servidor Edge se denota por $p^s(c, x)$, y la potencia consumida en el vBS se denota por $p^b(c, x)$.

Además, la observación de los indicadores de rendimiento es ruidosa en la práctica debido, por ejemplo, a perturbaciones aleatorias de los equipos o a periodos transitorios en la red. Es importante destacar que, es un requisito para los algoritmos de aprendizaje automático gestionar este ruido en las observaciones. En adelante, denotaremos por $d_t(c_t, x_t)$, $\rho_t(c_t, x_t)$, $p_t^s(c_t, x_t)$, y $p_t^b(c_t, x_t)$ las observaciones ruidosas de los indicadores de rendimiento en el periodo de tiempo t .

4 FORMULACIÓN DEL PROBLEMA

4.1. Caso de Uso 1

Comenzamos con el caso en el que el suministro de energía es escaso o, lo que es lo mismo, el operador desea reducir los costes de consumo de energía. Esto puede lograrse con una función objetivo escalarizada:

$$u(\omega_t, x_t) := r(\omega_t, x_t) - \delta \cdot B(P(\omega_t, x_t)),$$

donde $P(\omega_t, x_t)$ es el consumo de energía vBS asociado al par contexto-control (ω_t, x_t) , $B(\cdot)$ es una función smooth que modela el coste asociado al consumo de energía, y el parámetro δ determina la importancia relativa del coste de energía y el rendimiento alcanzado, y puede seleccionarse en función de las preferencias del operador. También utilizaremos $u_t(\omega_t, x_t)$ para denotar la realización de la función objetivo relacionada con las muestras del periodo t , $P_t(\omega_t, x_t)$ y $r_t(\omega_t, x_t)$. La selección de la función de coste es crucial aquí. En el caso más sencillo, puede ser una función lineal que asigne la potencia real consumida a un valor monetario (recompensa negativa). Pero también puede modelar situaciones en las que las políticas que superan un umbral de potencia deben evitarse debido a la regulación, las limitaciones de la batería, etc. Para capturar todos estos casos, proponemos utilizar una función sigmoidea parametrizada con parámetros de agudeza e inclinación a y b :

$$B(x) := \frac{1 + e^{ab}}{e^{ab}} \left(\frac{1}{1 + e^{-a(x-b)}} - \frac{1}{1 + e^{ab}} \right),$$

Cuando $a \rightarrow 0$, la función $B(\cdot)$ se aproxima a una función lineal, y cuando a crece [21] se aproxima a la función escalón, sin por ello inducir gradientes infinitos, lo cual dificultaría el trabajo de los algoritmos de aprendizaje automático.

Siguiendo el enfoque estándar en la optimización [22]–[23], utilizamos el *contextual regret* acumulado para evaluar el rendimiento de nuestro algoritmo. En concreto, definimos el contextual regret medio en el periodo T :

$$R_T := \sum_{t=1}^T \left(\max_{x' \in \mathcal{X}} u(\omega_t, x') - u(\omega_t, x_t) \right),$$

donde $\max_{x' \in \mathcal{X}} u(\omega_t, x')$ proporciona la mejor decisión para el periodo actual, que no podemos calcular en la práctica, ya que la función objetivo es desconocida. Nuestro objetivo, por tanto, es encontrar una secuencia de decisiones $\langle x_t \rangle_{t=1}^T$ del conjunto \mathcal{X} que garanticen un pseudo-regret medio asintóticamente sublineal, es decir, $\lim_{T \rightarrow \infty} E[R_T]/T = 0$, donde la esperanza se toma con respecto a las muestras ruidosas y el proceso de llegadas del contexto.

4.2. Caso de uso 2

Un problema diferente surge cuando el vBS funciona con un presupuesto de energía P_{max} estricto, por ejemplo, cuando se alimenta a través de Ethernet (PoE) o mediante una batería. En estos casos, el algoritmo de aprendizaje automático tiene que encontrar la configuración de máximo rendimiento que respete el máximo de energía disponible. Es importante destacar que el algoritmo de aprendizaje automático tiene que lograr este objetivo mediante el empleo de una exploración segura del espacio de configuración \mathcal{X} con el fin de satisfacer el umbral P_{max} en cualquier periodo, es decir, no sólo en la etapa final de operación o inferencia. Definimos el regret de la siguiente forma:

$$R_T^S := \sum_{t=1}^T \left(\max_{x' \in S_t(\omega_t)} r(\omega_t, x') - r(\omega_t, x_t) \right),$$

donde en este caso las decisiones se seleccionan del conjunto

$$S_t(\omega_t) = \{x \in \mathcal{X} \mid P(\omega_t, x) \leq P_{max}\}.$$

Nótese que en la definición del regret utilizamos directamente la función de recompensa r , ya que la potencia se considera ahora una restricción estricta. Nuestro objetivo es encontrar una secuencia $\langle x_t \rangle_{t=1}^T$, $x_t \in S_t(\omega_t)$, tal que $\lim_{T \rightarrow \infty} E[R_T^S]/T = 0$. Es importante destacar que los conjuntos $S_t(\omega_t)$, $\forall \omega_t$, son desconocidos inicialmente, ya que $P(\omega, x)$ también es desconocido, y por lo tanto necesitamos aprenderlos usando las medidas en tiempo real $P_t(\omega_t, x_t)$. Del mismo modo, sólo tenemos acceso a r_t y u_t , es decir, las mediciones ruidosas del período t , en lugar de las funciones reales r y u .

4.3. Caso de uso 3

El consumo de energía es uno de los principales componentes de los costes operativos de las redes móviles, y se espera que su impacto aumente aún más con el despliegue de los servicios AI/ML, que elevan aún más este peaje. Así lo han puesto de manifiesto diversos informes de vendedores, fabricantes y operadores [15]–[16].

Por lo tanto, nuestro objetivo es minimizar el consumo de energía de todo el sistema (vBS y servidor Edge) sujeto a las limitaciones de rendimiento del servicio. Dependiendo del factor de forma del vBS y de la configuración del servidor (es decir, modelo de GPU, placa base, etc.), la energía consumida por cada entidad puede tener un orden de magnitud diferente. Además, el coste asociado al consumo de energía puede variar en función del escenario. En escenarios regulares basados en celdas pequeñas, dicho coste puede estar relacionado con el precio de la electricidad, que puede variar entre el día y la noche en función de las tarifas establecidas por los proveedores de energía en cada país. En otros escenarios, como los basados en PoE o un vBS alimentado por energía solar, este coste puede reflejar la escasez del recurso energético para la RAN. Para capturar estos diferentes escenarios, definimos la siguiente función de coste:

$$u(c, x) = \delta_1 p^s(c, x) + \delta_2 p^b(c, x)$$

donde δ_1 y δ_2 son los costes de la potencia en el servidor de borde y el vBS, respectivamente, en unidades monetarias por watio (m.u./W).

Por otro lado, consideramos las restricciones de rendimiento a nivel de servicio, yendo un paso más allá de otros trabajos que consideran requisitos de rendimiento de nivel inferior (por ejemplo, [24]), como la velocidad de datos o el retardo.

El mapeo entre los pares contexto-acción y los indicadores de rendimiento a nivel de servicio es muy complejo y no existen modelos disponibles, como detallamos en los resultados experimentales de SORUS-RAN-A2.1. Por ello, los aprendemos a partir de observaciones. Para nuestro servicio de reconocimiento de objetos, consideramos dos restricciones: (i) un retardo máximo del servicio denotado por d^{max} , que está directamente relacionado con la velocidad de imágenes por segundo⁴ (número de imágenes por segundo) que el usuario va a procesar, y (ii) un mAP mínimo denotado por ρ^{min} que indica un límite inferior sobre la precisión del servicio en la detección de los objetos. Formulamos el problema de la siguiente manera:

$$\begin{aligned} \min_{\{x_t\}_{t=1}^T \in \mathcal{X}} \quad & \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T u_t(c_t, x_t) \\ \text{s.t.} \quad & d_t(c_t, x_t) \leq d^{max}, \quad \forall t \leq T \\ & \rho_t(x_t) \geq \rho^{min}, \quad \forall t \leq T. \end{aligned}$$

Obsérvese que las restricciones de servicio se satisfacen para el usuario que experimenta el peor servicio como $d(c, x) := \max_i D_i(c, x)$ y $\rho(c, x) := \min_i Q_i(c, x)$.

Cabe mencionar que estas funciones son desconocidas de antemano y dependen de la plataforma. Es decir, los valores de los indicadores de rendimiento pueden cambiar con el software que implementa el vBS, el hardware del servidor y el vBS, y el servicio que se ejecuta en el servidor Edge. En SORUS-RAN-A2.1 mostramos que las relaciones entre las políticas de control y los indicadores de rendimiento para un usuario no son triviales ni lineales. Con un mayor número de usuarios, estas relaciones pueden ser aún más complejas y mostrar un comportamiento diferente, lo que hace imprescindible el uso de algoritmos de aprendizaje automático.

Además, no existe una configuración óptima única, ya que depende del contexto. Esto hace que el problema sea muy complicado, ya que el conjunto de contextos es infinito y los contextos observados cambian con el tiempo. Para aprender la correspondencia entre el contexto y la configuración óptima, formulamos el problema como un contextual multi-armed bandit con restricciones.

Nos gustaría señalar que se pueden considerar otras formulaciones alternativas. Por ejemplo, podríamos considerar vBSs con restricciones de potencia o un presupuesto de potencia de computación Edge incluyendo los objetivos de consumo de energía como restricciones, al tiempo que se minimiza la latencia y se maximiza la mAP. La formulación de un problema de contextual

⁴ Consideramos un escenario con velocidad de fotogramas variable ya que algunos trabajos muestran que esto puede proporcionar beneficios sustanciales al rendimiento del servicio [25]–[27]. Sin embargo, también podemos considerar una velocidad de fotogramas fija como un caso particular de esta formulación.

bandit es una particularización de la conocida formulación del aprendizaje reforzado o RL con varias diferencias.

En primer lugar, la formulación RL considera que la distribución de probabilidad del estado/contexto en $t + 1$ depende del estado/contexto y de la acción seleccionada en t . Por el contrario, la formulación de contextual bandit considera que el contexto no depende de las acciones seleccionadas. Se puede observar que los contextos de nuestro problema (número de usuarios y calidad del canal) no pueden verse afectados por la configuración del sistema.

En segundo lugar, la formulación RL considera que la recompensa puede ser escasa o retardada, es decir, que las acciones seleccionadas afectarán a la recompensa un número arbitrario de pasos temporales más tarde. En algunos casos, la recompensa sólo se revela al final del episodio y el algoritmo debe distribuir el crédito de este resultado entre todas las acciones del episodio. En cambio, la formulación de contextual bandit considera que la recompensa es instantánea, es decir, que para un determinado par estado-acción en el tiempo t , podemos observar la recompensa asociada instantáneamente. Además, las decisiones tomadas en el tiempo t no repercutirán en futuros pasos temporales. Esta propiedad se cumple en nuestro problema debido a la escala temporal del proceso de toma de decisiones. Recordemos que nuestra solución opera en el non-RT RIC de la arquitectura O-RAN (escala temporal de segundos). Basándonos en esta escala temporal, la observación del rendimiento en cada momento t es independiente de los pasos temporales anteriores, ya que el sistema se vuelve estacionario.

Por último, cabe mencionar que el enfoque de contextual bandit no necesita estimar las recompensas acumuladas hasta el final del episodio como es necesario en RL (utilizando Aprendizaje por Diferencia Temporal, Métodos Monte Carlo, etc.). Estas simplificaciones en la formulación hacen que nuestra solución sea más sencilla y eficaz.

El contextual bandit $\pi_t(c_t; \mathcal{B}_{t-1}): \mathcal{C} \rightarrow \mathcal{X}$ es un algoritmo que asigna contextos a acciones, donde $\mathcal{B}_{t-1} = [c_1, x_1, u_1, d_1, \rho_1, \dots, c_{t-1}, x_{t-1}, u_{t-1}, d_{t-1}, \rho_{t-1}]$ es el conjunto de observaciones históricas en $t - 1$ (es decir, tripletas de contextos, acciones y métricas de rendimiento).

En cada periodo de tiempo t se observa un contexto c_t y se calcula una acción $x_t = \pi_t(c_t; \mathcal{B}_{t-1})$ y se aplica al sistema. Al final del periodo de tiempo t se miden los indicadores de rendimiento asociados al par (c_t, x_t) para actualizar \mathcal{B}_t y, en consecuencia, π_t . El contextual bandit π_t se actualiza secuencialmente con cada nueva medición del sistema hacia la solución del problema formulado anteriormente.

5 CONCLUSIONES

La virtualización, en general, reporta muchos beneficios en aspectos cruciales hoy en día como la flexibilidad, la capacidad de adaptación a los requerimientos de los usuarios, o la desvinculación del software de las plataformas hardware específicas que se estén utilizando. Sin embargo, para alcanzar una virtualización completa de estas redes y aprovechar al máximo los beneficios, es necesario afrontar retos importantes en elementos clave para conseguirlo dentro de esta nueva arquitectura, como son las estaciones base.

En este documento se han expuesto mecanismos basados en IA para lograr un compromiso entre flexibilidad y rendimiento en las denominadas vBSs y los servidores Edge asociados a las mismas. Se han propuesto tres casos de uso atendiendo a diferentes aspectos significativos como son: el

equilibrio entre flexibilidad y coste, la limitación de potencia máxima de la estación base virtualizada y la optimización conjunta de vBS y servicios de IA en el servidor Edge. Para generar los modelos de aprendizaje automático se han utilizado las métricas conocidas como: información de contexto, políticas de control y funciones de recompensa. Esto ha dado lugar a la creación de diferentes fórmulas que pueden ser utilizadas para modelar y asignar de forma eficiente, adaptativa y autónoma los recursos de acceso a la red en las vBSs, teniendo en cuenta las condiciones de la propia red, las necesidades de los usuarios y el consumo energético.

6 REFERENCIAS

- [1] "AT&T and Nokia Accelerate the Deployment of RAN Open Source," AT&T. Press Release, 2019. [Online]. Available: HTTPS://ABOUT.ATT.COM/STORY/2019/OPEN_SOURCE.HTML
- [2] I. Gomez-Miguel et al., "srsLTE: an Open-source Platform for LTE Evolution and Experimentation," in Proc. of ACM WinTech, 2016.
- [3] Nikaein, Navid, et al. "OpenAirInterface: A flexible platform for 5G research." ACM SIGCOMM Computer Communication Review 44.5 (2014): 33-38.
- [4] "Virtualized Radio Access Network: Architecture, Key Technologies and Benefits," Samsung. Technical Report, 2019.
- [5] "Open & Virtualized – The Future of Radio Access Network," NEC. White Paper, 2020.
- [6] O-RAN Alliance, "O-RAN-WG1-O-RAN Architecture Description - v01.00.00." Technical Specification, February 2020.
- [7] "Reimagining the End-To-End Mobile Network in the 5G Era," Cisco, Rakuten, Altiostar. White Paper, 2019.
- [8] C. Marquez et al., "How should i slice my network? a multi-service empirical evaluation of resource sharing efficiency," in Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, 2018, pp. 191–206.
- [9] J. Mendes et al., "Cellular access multi-tenancy through small-cell virtualization and common rf front-end sharing," Computer Communications, vol. 133, pp. 59–66, 2019.
- [10] Chatzopoulos, Dimitris, et al. "Mobile augmented reality survey: From where we are to where we go." Ieee Access 5 (2017): 6917-6950.
- [11] Microsoft, "Seeing AI," <HTTPS://WWW.MICROSOFT.COM/EN-US/AI/SEEING-AI>.
- [12] Taigman, Yaniv, et al. "Deepface: Closing the gap to human-level performance in face verification." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
- [13] Garcia-Saavedra, Andres, et al. "Joint optimization of Edge computing architectures and radio access networks." IEEE Journal on Selected Areas in Communications 36.11 (2018): 2433-2443.
- [14] GSMA - Future Networks, "Energy efficiency: An overview," 2019. [Online]. Available: <HTTPS://WWW.GSMA.COM/FUTURENETWORKS/WIKI/ENERGY-EFFICIENCY-2>
- [15] China Mobile Limited, "2020 Sustainability Report," white Paper, 2020.
- [16] GSMA Association, "5G energy efficiencies: green is the new black," white Paper, 2020.
- [17] O-RAN Alliance, "O-RAN-WG1-O-RAN Architecture Description (O-RAN.WG1.O-RAN-Architecture-Description-v04.00)," Technical Specification, March 2021.
- [18] Garcia-Saavedra, Andres, and Xavier Costa-Perez. "O-RAN: Disrupting the virtualized RAN ecosystem." IEEE Communications Standards Magazine 5.4 (2021): 96-103.
- [19] Mo, Jeonghoon, and Jean Walrand. "Fair end-to-end window-based congestion control." IEEE/ACM Transactions on networking 8.5 (2000): 556-567.
- [20] Everingham, Mark, et al. "The pascal visual object classes challenge: A retrospective." International journal of computer vision 111 (2015): 98-136.

- [21] Diez, Luis, et al. "LaSR: A supple multi-connectivity scheduler for multi-RAT OFDMA systems." *IEEE Transactions on Mobile Computing* 19.3 (2018): 624-639.
- [22] Krause, Andreas, and Cheng Ong. "Contextual gaussian process bandit optimization." *Advances in neural information processing systems* 24 (2011).
- [23] Sui, Yanan, et al. "Stagewise safe bayesian optimization with gaussian processes." *International conference on machine learning*. PMLR, 2018.
- [24] Ayala-Romero, Jose A., et al. "vrain: Deep learning based orchestration for computing and radio resources in vrans." *IEEE Transactions on Mobile Computing* 21.7 (2020): 2652-2670.
- [25] Inoue, Yusuke, Takatsugu Ono, and Koji Inoue. "Real-time frame-rate control for energy-efficient on-line object tracking." *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 101.12 (2018): 2297-2307.
- [26] Zhao, Yang, Qi Ni, and Ruoxin Zhou. "What factors influence the mobile health service adoption? A meta-analysis and the moderating role of age." *International Journal of Information Management* 43 (2018): 342-350.
- [27] Sodhro, Ali Hassan, et al. "Mobile edge computing based QoS optimization in medical healthcare applications." *International Journal of Information Management* 45 (2019): 308-318.