Plan de Recuperación, Transformación y Resiliencia

Financiado por la Unión Europea NextGenerationEU

uc3m | Universidad Carlos III de Madrid

SORUS

Validación y optimización conjunta de RIS y vRANs

SORUS-RAN A2.1

PERFILADO DE VRAN

Revisión	Autor	Fecha de entrega	Cambios
Versión 01	Jose Ayala Romero, Andrés García Saavedra	18/05/2023	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:

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

CONTENIDOS

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

LISTA DE ABREVIATURAS Y ACRÓNIMOS

Tabla 1. Lista de abreviaturas y acrónimos

Abreviatura	Explicación/Definición
Airtime	Tiempo de transmisión
AP	Average Precision
BBU	Unidad de Banda Base
BS	Base Station
BSR	Buffer State Report
CQI	Channel Quality Indicator
DL	Downlink
GPU	Graphics Processing Unit
IA	Inteligencia Artificial
mAP	mean Average Precision
MCS	Modulation and Coding Scheme
ML	Machine Learning
RAN	Radio Access Network
RAPL	Running Average Power Limit
RF	Radio Frecuencia
RIC	RAN Intelligent Controller
RU	Unidad de Radio
SCPI	Standard Commands for Programmable Instruments
SFF	Small Form Factor
SNR	Signal-to-Noise Ratio
TM	Modo de Transmisión
UE	User Equipment
UL	Uplink
vBS	virtualized Base Station
vRAN	virtualized Radio Access Network

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

LISTA DE FIGURAS

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

LISTA DE TABLAS

1 RESUMEN EJECUTIVO

Este documento corresponde al entregable A2.1 –Perfilado de vRAN– que se encuentra en la fase inicial de las tareas descritas en el pliego de condiciones técnicas. En este documento se definen los elementos utilizados en la implementación de dos testbed para la realización de diversos experimentos orientados a la obtención de datos de consumo de energía. Estos datos incluyen tanto a la CPU como a los dispositivos en su conjunto, pudiendo comparar los consumos energéticos de forma separada bajo los casos de uso planteados en el documento SORUS-RAN-A1.2. Esto ha permitido apreciar, de cara a la reducción del consumo de energía, que la plataforma hardware utilizada juega un papel fundamental en el consumo total, resultando esencial seleccionar la misma ajustada a las necesidades. A su vez, encontrar el balance correcto de MCS, airtime y resolución de las imágenes (en su caso), en cada situación es esencial para reducir el consumo energético sin perjudicar la experiencia del usuario final.

2 INTRODUCCIÓN

Las redes móviles de próxima generación están pensadas para hacer frente a una carga de tráfico creciente procedente de nuevas aplicaciones exigentes [1]. Un método prometedor para satisfacer estas necesidades es la densificación de la red: desplegar más estaciones base (BS) para reducir el tamaño de la celdas, ofrecer enlaces de alto rendimiento a los usuarios y reutilizar eficientemente el espectro inalámbrico. Esto cambiará drásticamente las redes celulares, que comprenderán órdenes de magnitud más BSs, con diferente tamaño y tecnología. De hecho, en la actualidad se está produciendo un rápido cambio de las estaciones base de hardware pesado a las estaciones base virtualizadas más pequeñas [2]. Ejemplos destacados son las plataformas de código abierto *OpenAirInterface [3]* y *srsRAN* [4], pero también iniciativas propietarias. Estas BSs implementan sus funciones mediante software, lo que ofrece una flexibilidad de gestión sin precedentes y admite su *virtualización* en hardware compartido [5].

Sin embargo, esta densificación y virtualización de la red tiene un coste para el medio ambiente y el presupuesto de los operadores. En particular, el consumo de energía representa entre el 15% y el 30% del OPEX de la red en los mercados en desarrollo, y el 70% de ese porcentaje se atribuye a las BS. Así pues, resulta evidente que el éxito del despliegue de las redes de próxima generación depende en gran medida de que seamos capaces de responder a las siguientes preguntas: *i)* ¿Cuánta energía consumen estas estaciones base virtualizadas (vBS)? *ii)* ¿Qué parámetros afectan a su consumo de energía? *iii)* ¿Cómo podemos reducir sus costes de energía? Algunas de estas preguntas ya se han planteado en el pasado, pero los estudios anteriores no ofrecen respuestas válidas para este contexto debido a dos razones principales.

En primer lugar, la virtualización de las BSs plantea una arquitectura completamente nueva. Mientras que las BSs tradicionales funcionan con hardware dedicado, las unidades de banda base (BBU) de las BS virtualizadas (vBS) se implementan en software que se ejecuta en procesadores de propósito general. Esto plantea dudas sobre su consumo de energía, ya que estudios anteriores han considerado equipos totalmente distintos. Además, mientras que en las BS tradicionales la potencia de transmisión es un factor de coste energético predominante, la densificación de la red implica una menor potencia de transmisión, lo que disminuye su importancia global. En este contexto, la potencia consumida en la BBU se convierte en un factor significativo del presupuesto energético global de las vBS, lo que motiva este trabajo experimental.

En segundo lugar, la carga computacional de la BBU cambia con varios parámetros que afectan de forma diferente al consumo de energía. En particular, los autores en [6] muestran que la carga computacional en la BBU presenta relaciones no lineales con la calidad del canal, el esquema de modulación y codificación (MCS) y la carga de tráfico; sin embargo, no está claro cómo estos pueden afectar al consumo de energía. En cambio, este comportamiento no aparece en las BS monolíticas o tradicionales. Por ejemplo, las mediciones realizadas en [7] muestran que el consumo de energía sólo varía hasta un 3% cuando la intensidad del tráfico pasa de un nivel sin carga a un nivel máximo.

En este entregable medimos tanto el rendimiento como el consumo de recursos en términos de energía para los tres casos de uso desarrollados en el entregable SORUS-RAN-A1.2.

3 METODOLOGÍA EXPERIMENTAL

3.1. Arquitectura de la plataforma experimental

Nuestra plataforma experimental está representada de forma esquemática en la Figura 1 y en su despliegue real en la Figura 2 y consta de la estación base virtual (vBS), el equipo de usuario (UE) y un medidor de potencia digital. La vBS consta de la cabeza de radio remota (RRH), para la que utilizamos la USRP B210 de Ettus Research, y la unidad de banda base (BBU) implementada en una plataforma de computación. Para evaluar el impacto del hardware, utilizamos dos PC de factor de forma pequeño (Intel NUC) y también dos servidores (ver **Error! Reference source not found.**) y todos ellos constan de procesadores desarrollados por Intel. El RRH y la BBU están conectados a través de un cable USB3.0, lo que significa que el RRH está totalmente alimentado por la BBU (sin fuente de alimentación externa). Para el UE, también utilizamos un Ettus Research USRP B210 y una plataforma informática de propósito general. La BBU y el UE se conectan directamente mediante cables SMA con atenuadores de RF de 20 dB. El cable de alimentación de la BBU está conectado al adaptador de medida GPM-001, que alimenta la BBU y permite al medidor de potencia GW-Instek GPM-8213 medir su consumo de energía en tiempo real. El UE comprende otro USRP B210 y una plataforma informática (no mostrada). De las posibles alternativas para la pila de protocolos 4G LTE de código abierto disponible, consideramos *srsRAN* porque en comparación con *OpenAirInterface* proporciona considerables ganancias en términos de tiempo de ejecución de la CPU, requisitos de memoria y estabilidad para un mayor ancho de banda [8]. Utilizamos la versión 19.12 de *srsRAN* y Ubuntu 18.04 en todas las plataformas de computación.

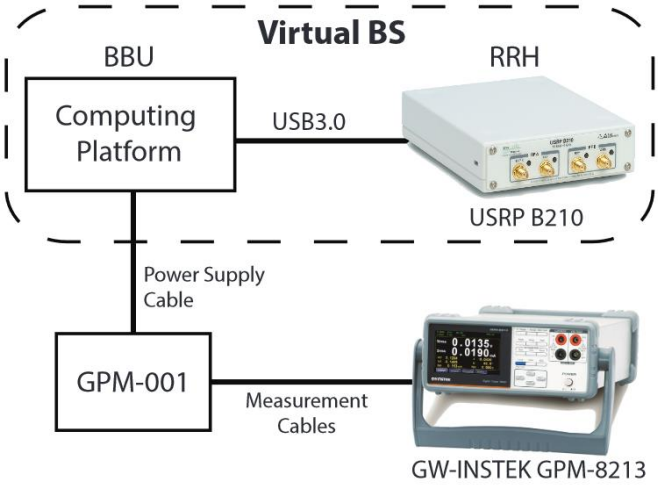

Figura 1. Esquema de conexión del vBS y el medidor de potencia

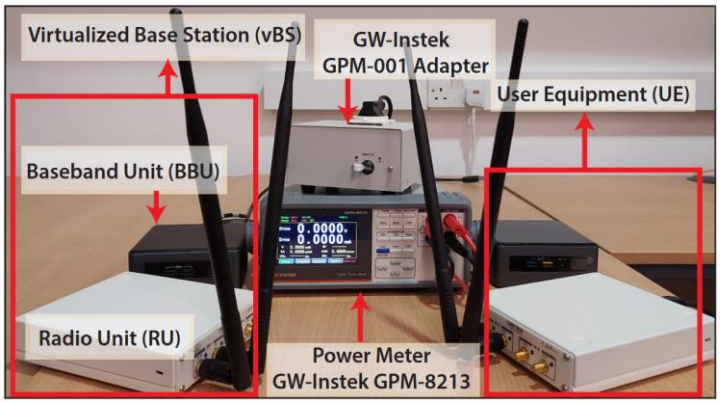

Figura 2. Foto del testbed inalámbrico que consta de una vBS (srsRAN) y un nodo que agrega el tráfico UE

Tabla 2. Plataformas informáticas consideradas en la evaluación experimental

Alias	Nombre comercial	CPU
NUC1	BOXNUC8I7BEH	i7-8559U @ 2.70GHz
NUC2	NUC7i7DNHE	i7-8650U @ 1.90GHz
Server1	Dell XPS 8900 Series	i7-6700 @ 3.40GHz
Server2	Dell Alienware Aurora R5	i7-9700 @ 3.00GHz

3.2. Medidas de potencia

Medimos el consumo de energía de vBS mediante software y hardware. Para el primero, utilizamos la funcionalidad Running Average Power Limit (RAPL) de Intel integrada en el kernel de Linux para medir la potencia de la CPU. RAPL proporciona varios contadores que indican información sobre el consumo de energía utilizando modelos de energía por software. Estos modelos estiman el consumo de energía utilizando contadores de rendimiento de hardware y modelos de I/O. Algunos trabajos han evaluado la precisión de las mediciones RAPL mostrando que, en la mayoría de los casos, coinciden con los valores reales de potencia [9]—[11]. Obtenemos las mediciones RAPL utilizando el programa Linux *turbostat*. También medimos la potencia vBS por hardware con el medidor GPM-8213 conectado como se ha explicado anteriormente. Nótese que, cuando medimos la potencia vía software, sólo estamos considerando la potencia de la CPU. Por otro lado, la medición vía hardware incluye el consumo de toda la plataforma (CPU, placa base, memoria RAM, etc.) más la parte de radio, ya que el RRH se alimenta a través del cable USB3.0 conectado a la BBU.

3.3. Metodología de medidas

Para la generación de nuestro conjunto de las medidas, establecemos la configuración del vBS y el UE durante un periodo de tiempo de 1 minuto. Tomamos muestras del consumo de energía de forma continua mediante software y hardware durante este periodo, y calculamos su media y varianza. Para cada muestra de nuestro conjunto de datos, configuramos el ancho de banda, el modo de transmisión (TM), la carga de tráfico en el enlace ascendente y descendente (uplink y downlink), la ganancia de transmisión en el UE (que afecta directamente a la SNR), el esquema de

modulación y codificación (MCS) y el tiempo de transmisión (airtime). La carga de tráfico se genera utilizando el programa *mgen*.

Utilizamos una versión modificada de *srsRAN* con la que cambiamos el MCS y el airtime a través de un socket TCP en tiempo de ejecución. La versión original de *srsRAN* sólo permite cambiar el MCS en el archivo de configuración (necesita reiniciar el vBS) y la configuración del airtime no está soportada. Nuestra versión personalizada también incluye otro socket TCP para obtener el rendimiento del vBS sobre la marcha utilizando el formato JSON. Un script Python selecciona la configuración de radio y recoge las mediciones de forma centralizada. Las medidas del medidor de potencia también se recogen desde Python utilizando la interfaz SCPI (Standard Commands for Programmable Instruments) a través de USB2.0. Finalmente, salvo que se indique lo contrario, los datos experimentales de las figuras mostradas en este entregable corresponden al dispositivo NUC1 configurado con un ancho de banda de 10MHz.

4 PERFILADO UPLINK

A continuación, caracterizamos el consumo de energía del enlace ascendente (uplink) en un vBS basada en CPUs a partir de un conjunto de medidas. Esta caracterización es realizada en función de la SNR y el airtime. Podemos calcular fácilmente el airtime a partir de la demanda de tráfico y el MCS. Por ejemplo, si la tasa de datos instantánea es de 20 Mbps y la demanda de tráfico es de 15 Mbps, el airtime es $a = 0{,}75$. Para este modelo, utilizamos el scheduler de radio por defecto de *srsRAN*, que selecciona el MCS para cada valor dado de SNR.

La Figura 3 (a) muestra las mediciones del consumo de energía de la CPU (puntos). Observamos que el consumo crece linealmente con la SNR. Esto se debe a que una mayor SNR permite el uso de un mayor MCS, lo que a su vez induce una mayor carga computacional de decodificación [6]. Recordemos que la asignación de SNR a MCS la realiza el scheduler de *srsRAN*. También observamos que a partir de cierto valor de SNR (aprox. 28 dBs) la potencia permanece constante. La razón es que a partir de este punto no se seleccionan MCS superiores y, por tanto, la carga computacional no aumenta más. La Figura 3 (a) también muestra el consumo de energía para diferentes valores de airtime (variable a en la leyenda), donde los valores punteados representan los valores experimentales y las líneas completas vienen dadas por una aproximación lineal a los datos. La reducción del airtime no sólo implica la reducción de los valores constantes de las curvas (la potencia consumida para la SNR más alta) sino también de la pendiente de las curvas.

La Figura 3 (b) muestra el impacto conjunto del airtime y la SNR en el throughput (tasa de datos). Observamos que, mientras que el throughput disminuye linealmente con el airtime (eje y), en el eje x depende del MCS asignado a cada valor de SNR. Aunque se puede conseguir el mismo throughput utilizando diferentes combinaciones de SNR y airtime, como muestra la Figura 3 (b), una SNR más alta y un airtime más bajo siempre reducirán el consumo de energía del vBS en este escenario. Sin embargo, conseguir una SNR alta puede ser costoso en términos de energía para el UE en algunos casos (aumento de su potencia de transmisión de enlace ascendente). Por lo tanto, el equipo de usuario puede decidir reducir su potencia de transmisión para ahorrar energía. Esta decisión implicará un aumento de la potencia consumida por el vBS, como muestran los experimentos. Tal y como revelan estos experimentos, en este escenario existen varios equilibrios no triviales.

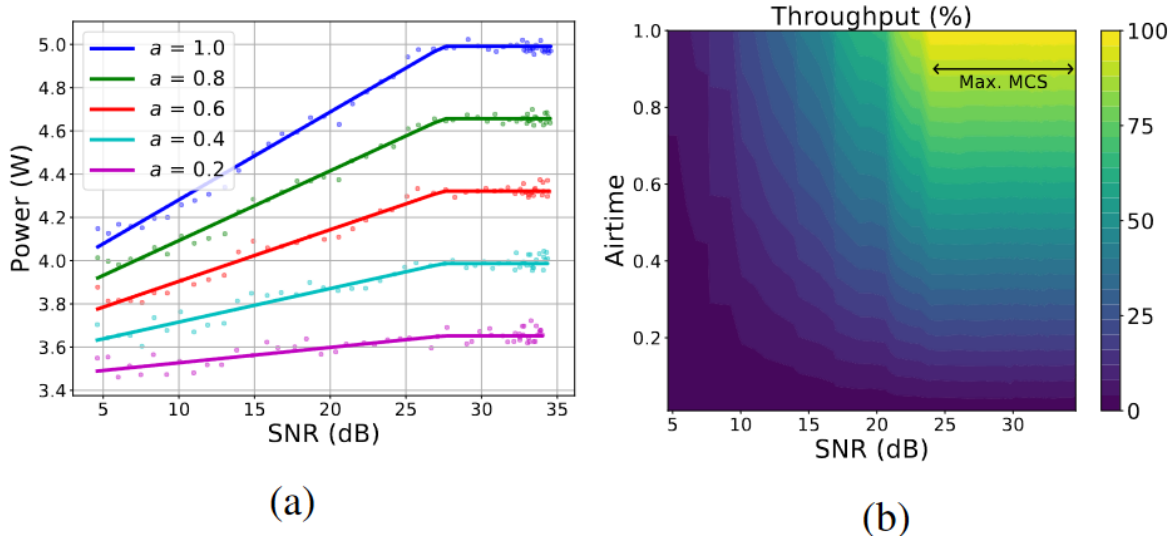

Figura 3. (a) Consumo de energía en el procesador en función de la SNR para distintos valores de airtime. (b) Porcentaje del throughput máximo en función del airtime y el SNR

En las figuras anteriores, estudiamos el impacto de la SNR y la carga de tráfico en el consumo de energía. Sin embargo, estos resultados estaban condicionados por el scheduler radio utilizado por *srsRAN*, que implementa una determinada regla para seleccionar el MCS en función de la calidad del canal medido (SNR). Si se rediseña el scheduler, lo que es posible en plataformas de código abierto, habría nuevas oportunidades para optimizar el consumo de energía.

El scheduler radio por defecto de *srsRAN* decide el MCS de cada usuario en función de su calidad de canal y depende de: *i)* el cálculo del Indicador de Calidad de Canal (CQI), y *ii)* el mapeo entre CQI y la coderate máximo. Mientras que este último está estandarizado ([12], Tabla 7.2.3-1), el cálculo del CQI no está definido en la especificación, ni está claro qué factores deben intervenir en él. *srsRAN* implementa el mapeo SNR a CQI de [13], y este aspecto está abierto a nuevas implementaciones. Es por ello que analizamos el consumo energético en función también de esta variable.

Sin embargo, no todos los MCS son viables para cualquier valor de SNR. Cuanto mayor es el MCS, menos ruido se tolera durante la descodificación. Además, la carga computacional de la descodificación aumenta cuando se reduce la SNR. Esto se debe a que el turbodecoder de la BBU necesita más iteraciones para valores de SNR más bajos, lo que implica un mayor tiempo de descodificación. Hemos realizado una serie separada de experimentos para medir este efecto, presentados en la Figura 4, que están en línea con trabajos anteriores, por ejemplo, [6].

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

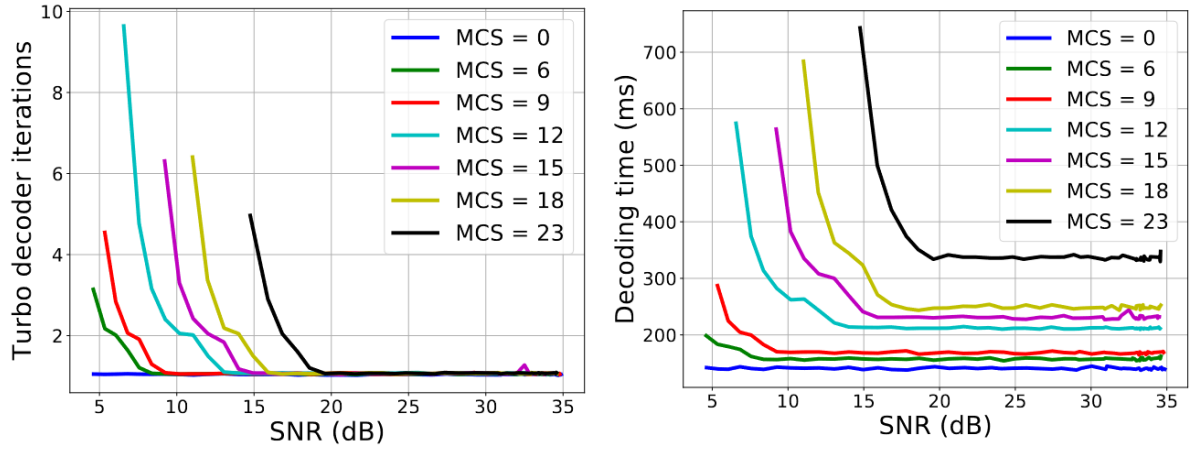

Figura 4. Iteraciones del turbodecoder y tiempo de decoding en función de la SNR para distintos valores de MCS.

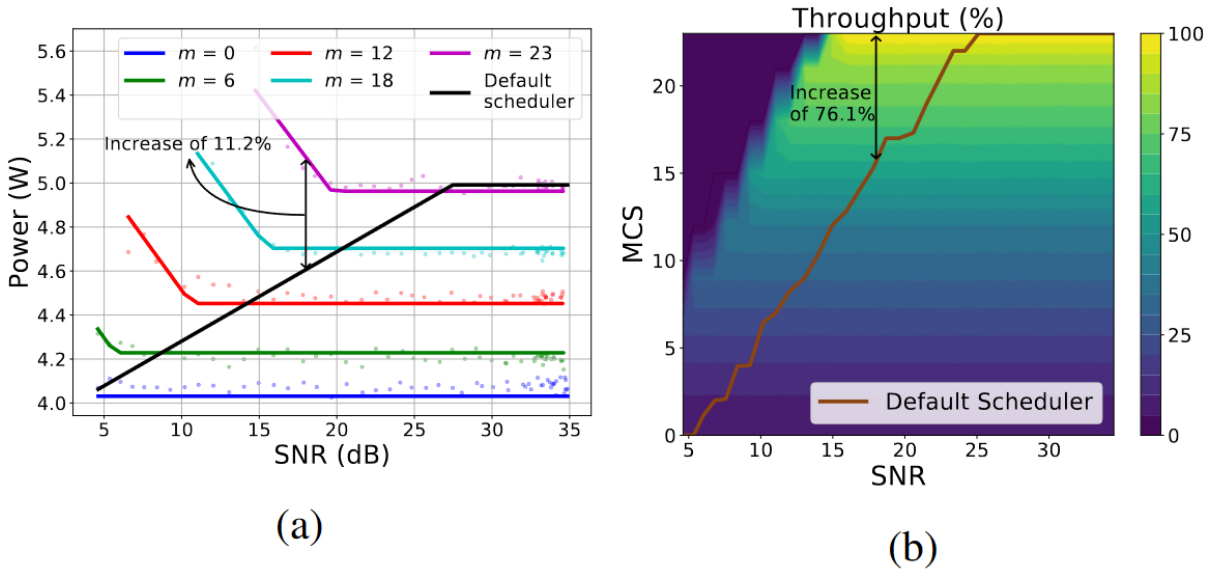

Figura 5. (a) Consumo de energía en el procesador en función de la SNR para diferentes MCS y airtime igual a 1. (b) Porcentaje del rendimiento máximo en función del MCS y la SNR para un airtime igual a 1

Figura 5-(a) muestra el consumo de energía de la CPU en función de SNR y MCS, para un airtime $a = 1$ (full buffer). Los puntos en la figura corresponden a los valores observados experimentalmente y las líneas completas vienen dadas por una aproximación lineal a las observaciones experimentales. En negro, mostramos el consumo de energía dado por el scheduler radio por defecto de srsRAN en el que se da la configuración MCS. También hemos incluido el consumo de energía del scheduler radio por defecto de *srsRAN* por comparación. Figura 5-(b) representa el throughput en función de SNR y MCS para full buffer. En marrón, mostramos la asignación MCS del scheduler radio por defecto srsLTE. Los valores cero de la esquina superior izquierda indican las combinaciones inviables de SNR y MCS, es decir, los puntos en los que la SNR no es lo suficientemente alta como para descodificar un MCS específico (error de descodificación). La línea marrón indica los valores de MCS seleccionados por el scheduler por defecto de *srsRAN* para una SNR dada. Observamos en la Figura 5 un compromiso entre potencia y rendimiento. Por ejemplo, en Figura 5-(a) y para SNR $c = 25$ dB, el scheduler de *srsRAN* consume 4.85 W, mientras

que si seleccionamos MCS $m = 23$ el consumo es de 5.12 W (11.2% de incremento de potencia). La Figura 5-(b) muestra el impacto de estas decisiones en el rendimiento que puede incrementarse en un 76,1% cuando seleccionamos el valor máximo de MCS en lugar del valor por defecto.

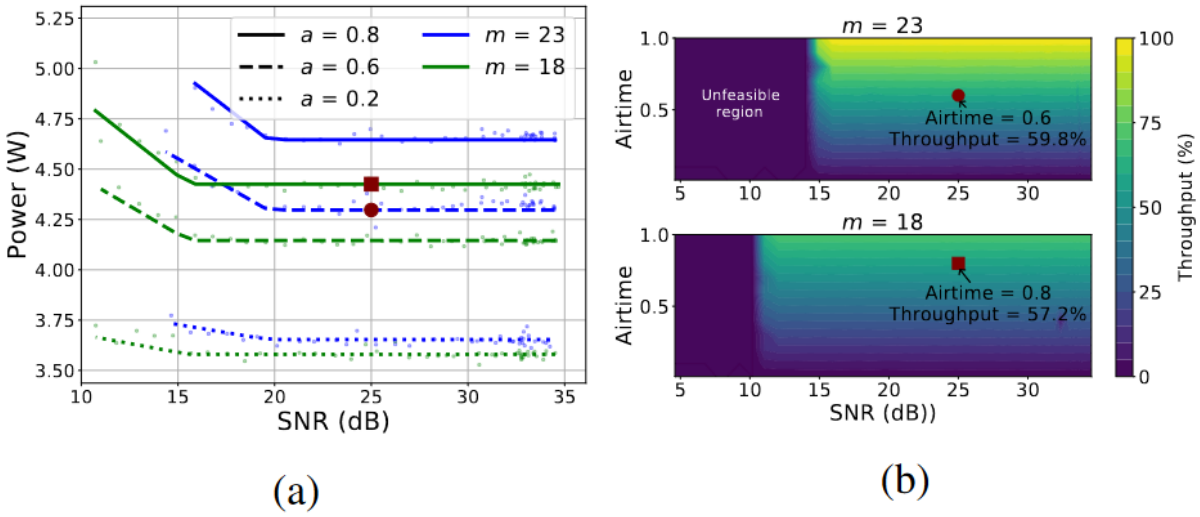

Figura 6. (a) Consumo de energía en el procesador en función de la SNR para distintos valores de MCS y airtime. (b) Porcentaje del throughput máximo en función del airtime y la SNR para los MCS seleccionados

La Figura 6-(a) muestra la dependencia del consumo de energía con el airtime para dos valores de MCS. La Figura 6-(b) muestra el efecto del airtime en el throughput para dos valores de MCS seleccionados. Los marcadores en marrón de esta figura indican dos configuraciones con un rendimiento similar. Estas dos configuraciones también están marcadas en la Figura 6-(a) en la que podemos observar que, aunque se consigue el mismo rendimiento, hay una pequeña diferencia en el consumo de energía. Esto es más frecuente cuando la diferencia entre los MCS es mayor. Esto indica que, cuando la calidad del canal es buena, el uso de MCSs más altos es más eficiente en términos de potencia. Sin embargo, si la SNR se reduce a 15 dB (siguiendo con el mismo ejemplo), esto cambia, es decir, la configuración con el MCS más bajo consume menos energía con el mismo rendimiento. Esto demuestra una vez más la relación no lineal entre la SNR y la potencia consumida en la BBU, que puede aprovecharse adecuadamente en función de las prioridades de la red en cuanto a rendimiento o costes.

Las dificultades de orquestación de vBS se ven agravadas por la plenitud de opciones de configuración que ofrecen estas estaciones base. La Figura 7 (a), por ejemplo, presenta combinaciones de valores de MCS y airtime (porcentaje de subtramas utilizadas) que logran el mismo throughput en el UL. Las configuraciones con mayor MCS (y, por tanto, menor airtime) reducen la potencia en un 38%. Sin embargo, esta relación es *no monótona*, ya que también hemos medido una mayor potencia cuando aumenta el MCS y la SNR es relativamente baja. Este último efecto se debe al rápido aumento de la carga computacional (véase la Figura 4). Por otro lado, las configuraciones 6 a 8 tienen el mismo consumo de energía, pero aun así difieren, ya que la configuración 8 implica menos tiempo de emisión y, por tanto, puede dar servicio a más usuarios, mientras que la configuración 6 es más resistente al ruido. Estas decisiones son tomadas por el

scheduler radio de la vBS[1] que selecciona el MCS y el airtime basándose en la SNR medida (contexto). Para este experimento, hemos modificado adecuadamente el scheduler radio de srsLTE con el fin de soportar diferentes valores de airtime. La Figura 7 (b) muestra el consumo de energía en función del MCS y del airtime para transmisiones UL. Observamos que ambos parámetros tienen un impacto suave en el consumo de energía.

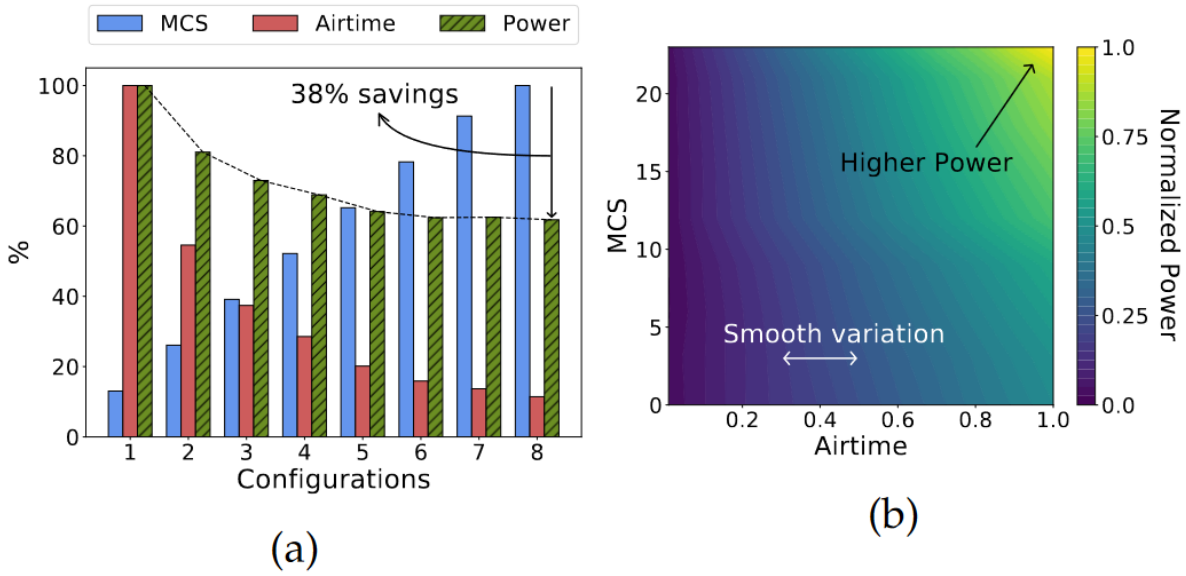

Figura 7. (a) Ocho configuraciones de MCS y airtime que ofrecen 2,6Mbps en UL, y la potencia respectiva (potencia del modo inactivo restada). (b) Consumo de energía normalizado de la BBU sobre la línea base, para transmisiones UL en full buffer y valores altos de SNR, en función de MCS y tiempo aire

La Figura 8 muestra el consumo total de energía y el consumo de energía de la CPU para 4 plataformas de computación y 3 valores de ancho de banda cuando el vBS funciona con transmisiones full buffer y SNR alta. Consideramos dos arquitecturas: *i)* dos PC de factor pequeño (NUC1 y NUC2) con un consumo de energía previsiblemente reducido y diversos procesadores con diferentes requisitos energéticos, y *ii)* dos servidores de propósito general (Server1 y Server2).

En primer lugar, observamos que el consumo de energía aumenta hasta 460,86% cuando cambiamos de un PC de factor pequeño a un servidor de propósito general. En segundo lugar, la selección del ancho de banda de la vBS es de suma importancia y debe ajustarse a los requisitos de la red para evitar no sólo el derroche de energía, sino también el uso ineficiente de los recursos radioeléctricos. Observamos un aumento medio del 20,27% cuando pasamos de 3 MHz a 10 MHz. En tercer lugar, incluso utilizando una plataforma con el mismo factor de forma encontramos diferencias importantes en el consumo de CPU y de energía total, por ejemplo, NUC2 consume un 86,93% más que NUC1 en media. Por último, observamos que la parte de la energía consumida por la CPU con respecto a la energía total es muy notable y oscila entre el 28,96% y el 49,69%.

[1] Por ejemplo, el scheduler de nuestra plataforma experimental selecciona el máximo MCS para una SNR dada y reduce el tiempo de emisión cuando el tráfico UL es inferior a la capacidad del enlace; pero para el tráfico DL selecciona MCS más bajos para que la comunicación sea más robusta, pero esto aumenta el consumo de energía.

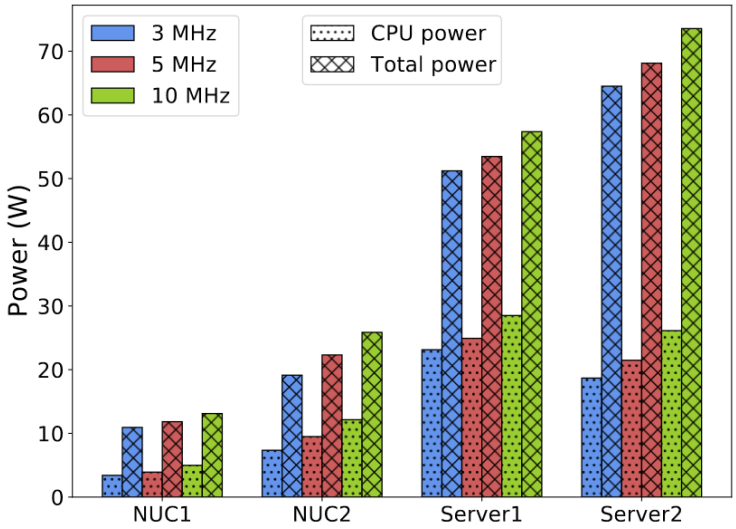

Figura 8. Comparación del consumo de energía para 3 anchos de banda y varias plataformas de computación, utilizando una SNR alta y transmisiones full buffer

5 PERFILADO CONJUNTO UPLINK Y DOWNLINK

Nuestro primer hallazgo es que el consumo de energía asociado con el procesamiento de la BBU es comparable a la potencia de transmisión de la cadena de radio frecuencia (RF). Este resultado es coherente con estudios anteriores; por ejemplo, [14] estimó que el 40% del consumo de energía de una femtocelda se debe a su BBU. En detalle, la Figura 9 (a) disecciona el consumo de energía de una vBS desplegada en un PC de factor de forma pequeño (SFF), y presenta los diferentes componentes de energía derivados de la CPU de la BBU[2]; la plataforma de computación de la BBU excepto las CPU; y la unidad de radio (RU) que se despliega sobre una radio definida por software USRP. Recordemos que los componentes utilizados son SF PC 1: Intel NUC i7-8559U@2.70GHz; SF PC 2: Intel NUC i7-8650U@1.90GHz; Servidor 1: Dell XPS 8900 i7-6700@3.40GHz; Servidor 2: Dell Aurora R5 i7-9700@3.00GHz. Para tener una visión completa, medimos el consumo de energía en cuatro escenarios diferentes: *(i)* el vBS no está desplegada (escenario base), *(ii)* el vBS está desplegado con un usuario inactivo conectado (vBS idle), *(iii)* el vBS está transmitiendo 20Mbps de tráfico de enlace descendente o dowlink (DL), y *(iv)* el usuario está transmitiendo 20Mbps de tráfico de enlace ascendente o uplink (UL) al vBS.

Excluyendo el escenario de referencia, el consumo de energía de la CPU es, de media, un 29% mayor que el consumo de energía de la RU; mientras que la potencia total de la BBU la supera en un 175% (208% con carga UL completa). Curiosamente, estas cifras dependen de la plataforma que aloje la BBU. En concreto, la Figura 9 (b) muestra el consumo de la BBU con respecto a la línea de base para varias plataformas[3]. Comparamos la potencia consumida por la BBU en estado de reposo y cuando funciona en full buffer para uplink y downlink, y restamos la potencia de referencia. De hecho, el

[2] Utilizamos la función Running Average Power Limit de Intel integrada en el kernel de Linux para el consumo de energía de la CPU.

[3] Los Small Factor PCs (SF PC) consumen menos energía que los servidores, que pueden alojar más vBS y, por tanto, consumen menos energía/usuario.

consumo de energía cambia significativamente, y también se ve afectado por el ancho de banda del vBS, otro parámetro configurable de las estaciones base virtualizadas.

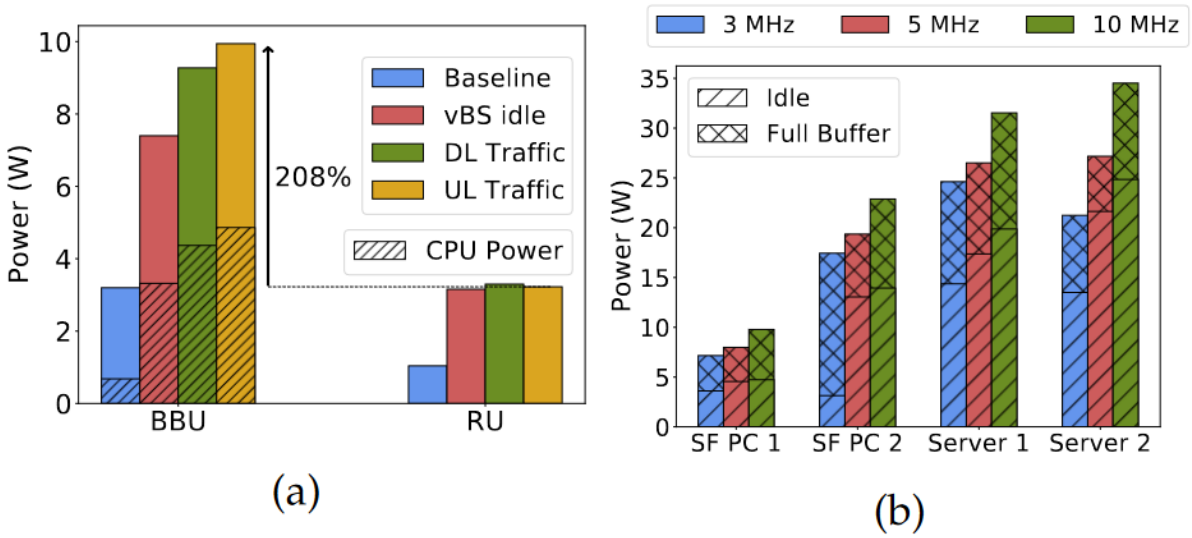

Figura 9. (a) Comparación del consumo de energía en la BBU, CPU de la BBU y la RU (un SDR USRP), con tráfico DL y UL de 20 Mbps. (b) Potencia consumida con respecto a la línea de base para diferentes anchos de banda de radio y plataformas hardware

Por último, la Figura 10 muestra el consumo de energía de la BBU cuando el tráfico DL y UL se procesa por separado y simultáneamente (UL+DL), para una SNR alta y varios valores de MCS. Los resultados han sido representados para tres casos: únicamente tráfico DL, únicamente tráfico UL, y tráfico combinado de DL y UL. Se observa que la potencia conjunta no es la suma total de los componentes separados. Por ejemplo, para MCS 15, el procesamiento concurrente DL y UL consume sólo un 7,5% más que el procesamiento sólo UL (y un 26% sobre el procesamiento sólo DL). Esto se debe a que hay factores de consumo de energía comunes en ambos flujos. Esto, a su vez, dificulta la predicción del consumo energético global del vBS, dado que el DL y el UL pueden configurarse por separado. También hay que tener en cuenta que los costes de UL son mayores y más volátiles que los de DL, ya que la descodificación es más exigente que la codificación desde el punto de vista computacional.

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

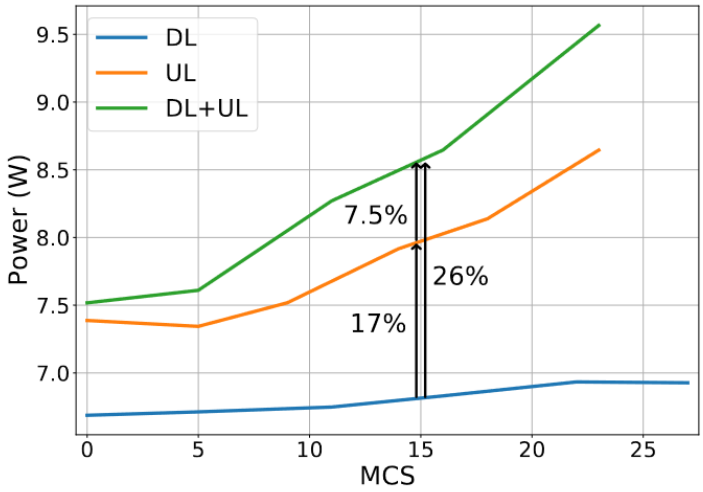

Figura 10. Impacto del MCS en el consumo de energía de la BBU con una SNR alta

6 PERFILADO APLICACIÓN IA EN EL BORDE DE LA RED

6.1. Arquitectura de la plataforma experimental

Para este caso de uso (caso de uso 3 en SORUS-RAN-A1.2), la plataforma experimental es ligeramente diferente a los casos anteriores (ver Figura 11). La plataforma experimental consta de una estación base virtualizada (vBS) LTE compatible con 3GPP R10, un equipo de usuario (UE) que genera peticiones de servicio a través de la vBS a un conocido servicio de reconocimiento de objetos y un servidor comercial con una GPU NVIDIA que ejecuta el servicio. Cada solicitud consiste en una imagen con un número variable de objetos del dataset COCO [15]. Las imágenes se envían al servicio a través del canal de subida (UL) de la interfaz LTE, y el servicio devuelve al usuario un conjunto de cuadro delimitadores y de etiquetas de clasificación para cada objeto identificado en la imagen. Esta información se envía a través del canal de enlace descendente (DL) de la interfaz LTE. Cada medición mostrada como un punto en las figuras mostradas a continuación es la media de 150 imágenes.

A continuación, analizamos los efectos de las diferentes políticas de configuración e indicadores de rendimiento: *(i)* la QoS experimentada por los usuarios; *(ii)* la energía asociada al servicio; y *(iii)* la energía de coste de la red móvil. Tanto las políticas de configuración como los indicadores de rendimiento han sido definidos en el entregable SORUS-RAN-A1.2. Sin embargo, se van a volver a definir en este entregable brevemente para que sea autocontenido. Para una explicación más detallada, ver SORUS-RAN-A1.2.

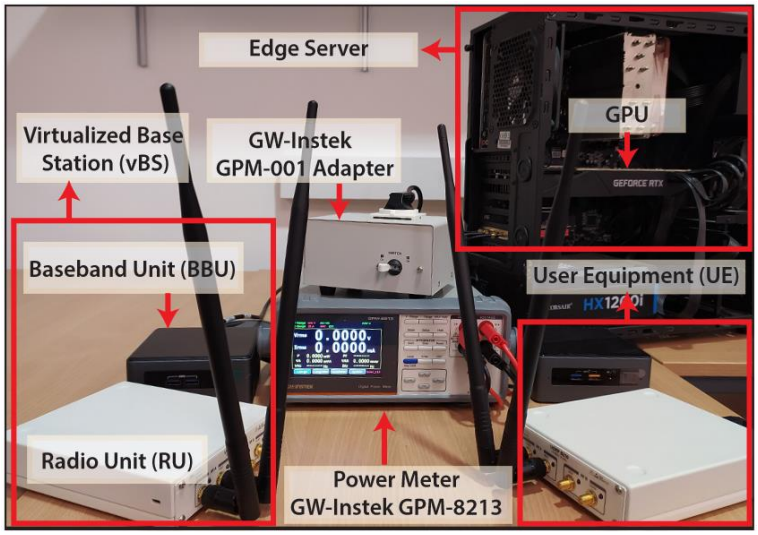

Figura 11. Plataforma experimental para el perfilado de servicios en el borde de la red

6.2. Perfilado

Comenzamos analizando dos métricas de interés para la QoS: el rendimiento del servicio para reconocer objetos y el tiempo de servicio, introducidos formalmente en Indicador de rendimiento 1 y 2, respectivamente.

Indicador de rendimiento 1 (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.

Indicador de rendimiento 2 (Precisión media (mAP)): La precisión del servicio se cuantifica mediante la Precisión Media o mean Average Precision (mAP). Para más detealles ver [16] y SORUS-RAN-A1.2.

Según nuestras mediciones, la característica más relevante que afecta al mAP es la resolución de imagen, véase Política de configuración 1.

Política de control 1 (Resolución de imagen): Esta política 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.

Lo ilustramos en la Figura 12, que muestra la relación entre tiempo de servicio (service delay en las figuras) y mAP para las imágenes del conjunto de datos COCO codificadas con diferentes resoluciones. El resto de políticas de configuración (descritas más adelante) son fijas. Los resultados revelan relaciones interesantes y cuantificables: *(i)* Las imágenes de mayor resolución llevan más píxeles codificados en una mayor cantidad de datos; por tanto, incurren en un mayor retardo debido al mayor tiempo de transmisión a través de la interfaz de radio. *(ii)* Las imágenes de menor resolución hacen que el servicio ofrezca un menor rendimiento mAP porque llevan menos información para el motor de detección de objetos. En concreto, medimos una mejora del retardo del 72% a expensas de una reducción de la precisión que oscila entre el 10% y el 50%.

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

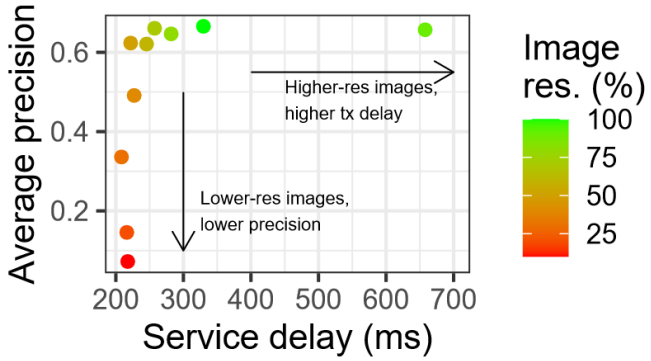

Figura 12. Precisión media (mAP) vs. Tiempo de servicio para imágenes con diferentes resoluciones

También existe una relación interesante que aparece de forma natural en muchos problemas de control de recursos, entre la calidad de servicio y los costes energéticos globales. Para explorar esto, introducimos una política que gobierna la asignación de recursos de radio (Política de control 2); y una métrica adicional que evalúa parte del coste mencionado: el consumo de energía del servidor (Indicador de rendimiento 3).

Política de control 2 (Radio airtime): Esta política de radio impone una restricción a los recursos de radio (ciclo de trabajo) que el vBS asigna al tráfico de servicio.

Indicador de rendimiento 3 (Consumo de potencia en el servidor): Coste energético asociado a la carga computacional de las peticiones del servicio, que está dominado por el consumo energético de la GPU.

La Figura 13 representa el tiempo de servicio vs el consumo de energía del servidor, para diferentes políticas de radio airtime y resoluciones de imagen. Como antes, las imágenes de mayor resolución aumentan el retardo del servicio debido al mayor tiempo de transmisión de las solicitudes. Ahora observamos que esto ocurre independientemente de la configuración de la política de radio. Sin embargo, la política de radio también tiene un impacto importante en el tiempo de servicio. Esto era de esperar, ya que un menor tiempo de emisión implica un menor uso de los recursos de radio, lo que aumenta aún más el tiempo de transmisión de las solicitudes en la interfaz de radio. En concreto, nuestros experimentos muestran que un aumento del 80 % en el tiempo de antena mejora el retardo entre un 65 % y un 80 %. En cuanto al consumo de energía del servidor, las imágenes de menor resolución y las asignaciones de recursos radioeléctricos inferiores aumentan este coste. En concreto, se produce un aumento del 56% en el consumo de energía para un aumento del 80% en el airtime de radio; se alcanza un aumento similar cuando se produce un aumento del 75% en la resolución de la imagen. Esto se debe a que el aumento de los recursos de radio permite al usuario enviar una mayor tasa de peticiones de forma similar a como lo hacen las imágenes de baja resolución, lo que en última instancia aumenta la carga de trabajo asignada a los recursos del servicio (la GPU, en este caso).

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

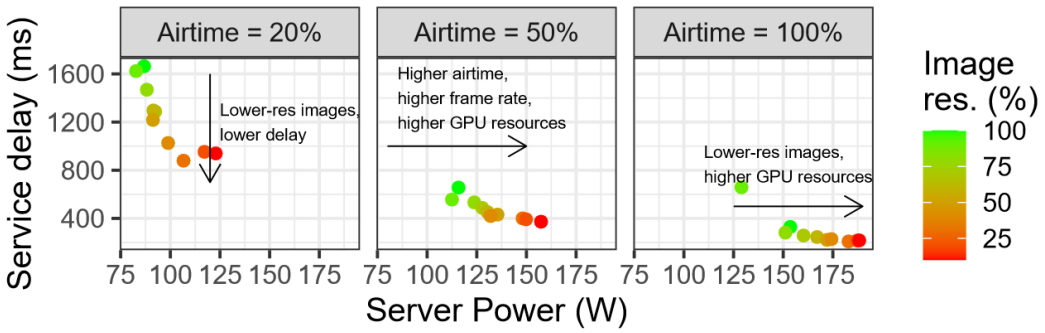

Figura 13. Retraso del servicio vs. consumo de energía del servidor para imágenes con diferentes resoluciones y políticas de radio

A continuación, estudiamos el impacto de las políticas de asignación de computación sobre la QoS. Para ello, definimos una política de control adicional.

Política de control 3 (Velocidad de la GPU): La política del servidor es un límite de potencia de la GPU que adapta la velocidad de procesamiento de una GPU (o un grupo de GPUs).

En nuestra configuración experimental, la velocidad de la GPU puede ajustarse a través de un parámetro de configuración disponible en los controladores de la GPU NVIDIA. La Figura 14 muestra el retardo del servicio y el consumo de energía del servidor para varias configuraciones de resolución de imagen. Ahora fijamos el tiempo de emisión en 100% y variamos la política de asignación de recursos de computación. Una mayor cantidad de recursos de computación aumenta el consumo de energía del servidor, ya que estamos relajando el límite de potencia impuesto a la GPU. Observamos que las imágenes de baja resolución contribuyen a aumentar el consumo de energía del servidor, ya que la tasa de peticiones también crece. Sin embargo, es interesante observar que las imágenes de mayor resolución alivian el trabajo de la GPU, como demuestra la Figura 14 (parte inferior), que muestra el retardo asociado únicamente a las tareas de la GPU. Con todo, a pesar de esta mejora en el retardo de la GPU, predomina el correspondiente aumento en el retardo de transmisión cuando se utilizan imágenes de mayor resolución. Es importante observar que, si bien esto es cierto en nuestra plataforma experimental, bien podría ser diferente para diversos despliegues (por ejemplo, una GPU más eficiente energéticamente, o una red de acceso radioeléctrico de mayor ancho de banda). Esto motiva la necesidad de algoritmos de aprendizaje automático que se adapten a los distintos despliegues.

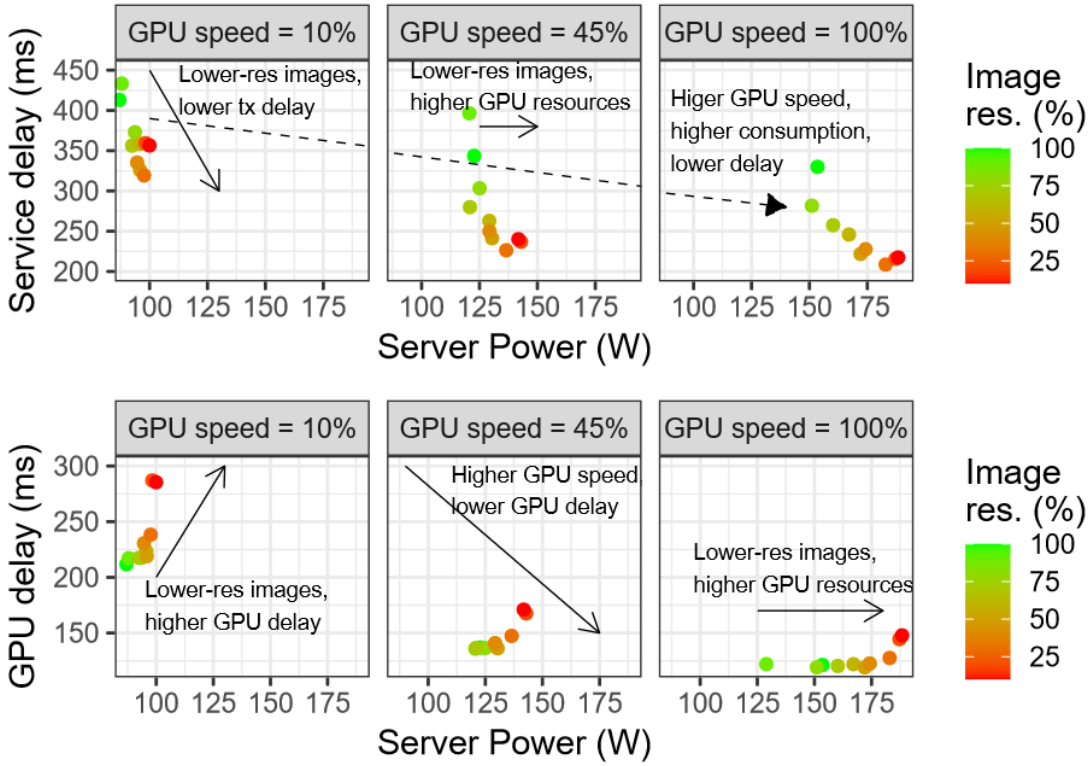

Figura 14. Tiempo de servicio vs. consumo de energía del servidor para imágenes con diferentes resoluciones y velocidades de GPU

La relación entre el tiempo del servicio y el consumo de energía del servidor también se observa en otras métricas de rendimiento, como el mAP. Para evaluar esto, la Figura 15 muestra el mAP conseguido por el servicio en función del consumo de energía del servidor para varias resoluciones de imagen. Los recursos radioeléctricos y computacionales se asignan de forma que se minimice el tiempo de servicio. Los resultados confirman que el coste del servicio depende del mAP. Sin embargo, es importante señalar que la relación con el mAP es sustancialmente distinta de la que existe con el tiempo del servicio. En este caso, un mayor rendimiento del mAP requiere en realidad un menor consumo de energía del servidor. La razón estriba en que las imágenes de mayor resolución (que proporcionan un mAP más alto) facilitan la detección de objetos y, por tanto, requieren menos recursos de computación (véase la Figura 14 (abajo)).

Por último, los costes asociados al operador de red están necesariamente condicionados por la cantidad de recursos radioeléctricos invertidos en la canalización del servicio. Para analizar esto, introducimos una política adicional, motivada por trabajos previos como [17] en el contexto de las RAN virtualizadas, que se define como Política de control 4, e Indicador de rendimiento 4, que refleja parte de los costes operativos del operador de red.

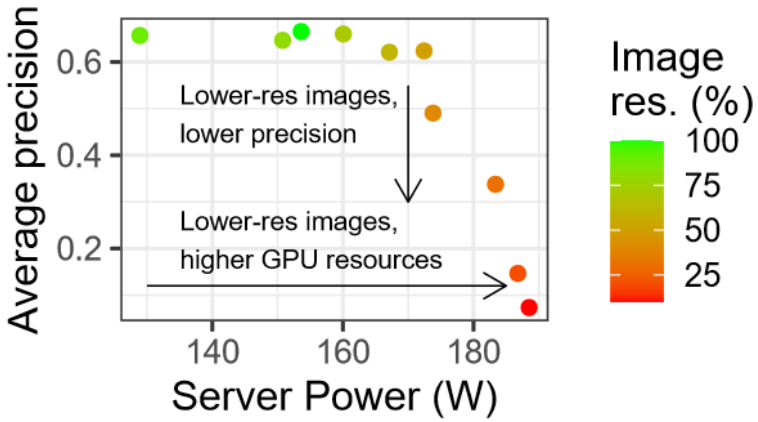

Figura 15. Precisión media (mAP) vs. consumo de energía del servidor para imágenes con distintas resoluciones

Política de control 4 (Radio MCS): Esta política impone una restricción al MCS máximo elegible por la vBS.

Indicador de rendimiento 3 (Potencia consumida por la vBS): Consumo de energía asociado al procesamiento de la unidad de banda base en un entorno RAN virtualizado.

Para analizarlos, representamos en la Figura 16 el consumo de energía medido en la BBU del vBS para distintas políticas de airtime, MCS y resoluciones de imagen. En primer lugar, observamos que las imágenes de menor resolución consumen menos recursos de radio y, por tanto, menos energía del vBS. En segundo lugar, el uso de mayores recursos radioeléctricos (airtime) induce en realidad mayores costes de energía porque permite al usuario transmitir imágenes a mayor velocidad. Por último, y quizás sorprendentemente, las políticas de MCS más elevadas provocan un menor consumo de energía de la vBS. La razón es que la carga de datos en la BS es relativamente baja en comparación con el ancho de banda disponible en la vBS; por ejemplo, las imágenes de mayor resolución con un 100 % de tiempo aire generan hasta 2,8 Mb/s, frente a una capacidad de unos 50 Mb/s (SISO LTE @ 20MHz de ancho de banda). En este escenario, a pesar de que las subtramas LTE moduladas con mayor MCS incurren en un mayor consumo instantáneo de energía, procesan la carga más rápidamente, lo que compensa en términos de consumo de energía a largo plazo.

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810

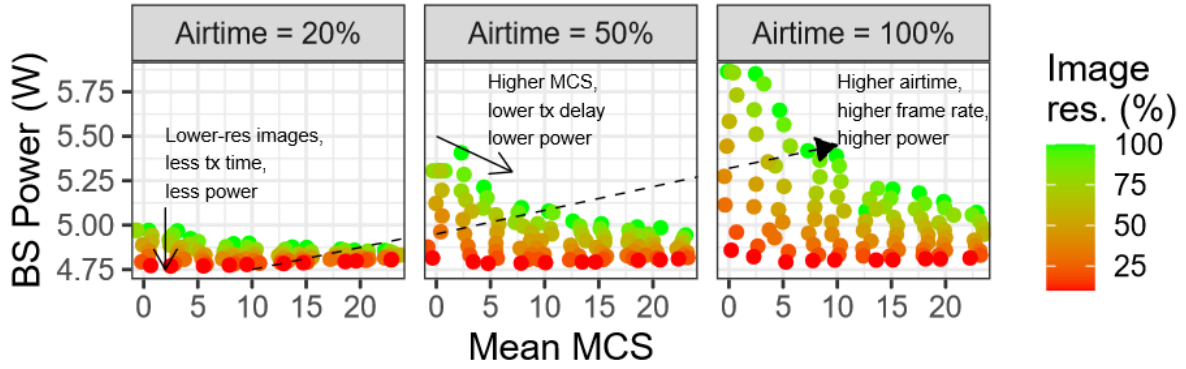

Figura 16. Consumo de energía de la vBS vs. políticas de radio para imágenes con diferentes resoluciones

A partir de estos resultados y desde el punto de vista de la vBS, no hay razón para utilizar un MCS inferior al máximo posible. Sin embargo, esto depende también de la carga de la red, que puede ser muy diferente para, por ejemplo, múltiples usuarios u otros servicios. Para demostrarlo, emulamos un escenario con 10 veces más carga, y presentamos el mismo gráfico en la Figura 17. A diferencia de ahora, observamos que la política de MCS tiene un impacto negativo en el consumo de energía de la vBS para las imágenes de mayor resolución, mientras que las imágenes de menor resolución provocan un menor consumo de energía para las políticas de MCS más altas. Esto motiva la necesidad de algoritmos de aprendizaje que adapten el sistema a los requisitos del servicio.

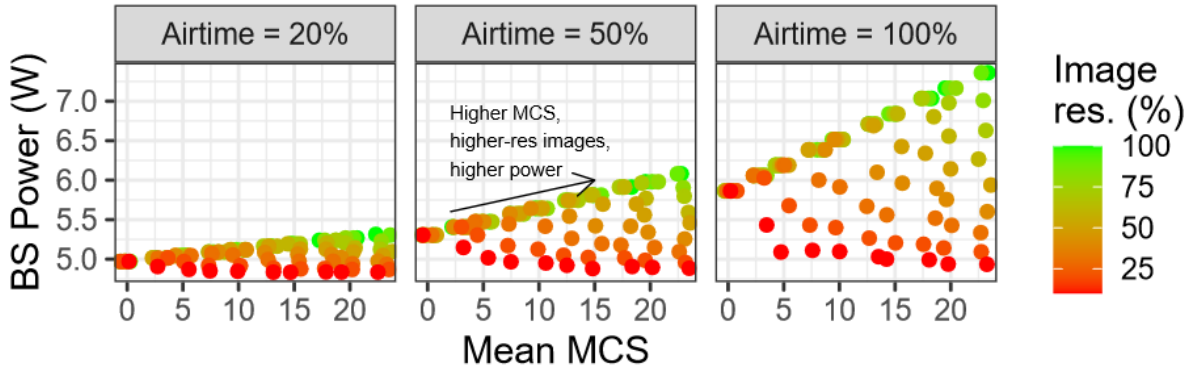

Figura 17. Consumo de energía de la BS vs. políticas de radio para imágenes con diferentes resoluciones y 10 veces mayor carga

7 CONCLUSIONES

El consumo energético es un factor fundamental al que se le debe prestar especial atención para conseguir un desarrollo estable y una implantación efectiva de vRAN. La finalidad del trabajo descrito en este documento es la de cuantificar las diferencias de consumo energético existentes en los distintos casos de uso propuestos en el documento SORUS-RAN-A1.2, así como analizar los resultados de las pruebas realizadas y estudiar posibles puntos de mejora.

Para realizar estas medidas de consumo de energía relacionada con los casos de uso 1 y 2, se ha dispuesto un testbed compuesto por una RRH, conectado a una BBU, en este caso un Intel NUC, dos

servidores Intel y un UE . Las mediciones de consumo de energía se han realizado simultáneamente por software y por hardware. Para la medición software se ha utilizado la funcionalidad implementada por Intel RAPL en el kernel de Linux con la que se han obtenido valores únicamente del consumo de la CPU. En cambio, la medición por hardware, se ha efectuado mediante instrumentos de medida externos, proporcionando datos del consumo global de los dispositivos, algo que ha resultado útil para realizar una comparación y averiguar cuáles son los elementos más demandantes de energía bajo cada una de las situaciones.

En particular, para el caso de uso 3, se ha dispuesto de un testbed similar al anterior, con la salvedad de incluir un servidor comercial, que dispone de una GPU NVIDIA, que es el encargado de ofrecer un servicio de procesamiento de imágenes. Se ha podido cuantificar el consumo de energía bajo distintos escenarios, lo que ha permitido advertir que es necesario desarrollar algoritmos de aprendizaje que adapten la resolución de las imágenes, el MCS y airtime para optimizar el consumo sin perjudicar la QoS. Además, como norma general, la correcta selección del hardware a utilizar para cada una de las tareas es imprescindible para optimizar el consumo de energía, ya que, según los experimentos llevados a cabo, la diferencia de consumo total puede llegar a ser siete veces mayor utilizando el servidor de mayor potencia de cómputo en lugar del servidor SFF de menor consumo.

8 REFERENCIAS

[1] Alliance, N. G. M. N. "5G white paper." Next generation mobile networks, white paper 1.2015 (2015).

[2] Afolabi, Ibrahim, et al. "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions." IEEE Communications Surveys & Tutorials 20.3 (2018): 2429-2453.

[3] Nikaein, Navid, et al. "OpenAirInterface: A flexible platform for 5G research." ACM SIGCOMM Computer Communication Review 44.5 (2014): 33-38.

[4] Gomez-Miguelez, Ismael, et al. "srsLTE: An open-source platform for LTE evolution and experimentation." Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization. 2016.

[5] Rost, Peter, et al. "Mobile network architecture evolution toward 5G." IEEE Communications Magazine 54.5 (2016): 84-91.

[6] Rost, Peter, Salvatore Talarico, and Matthew C. Valenti. "The complexity–rate tradeoff of centralized radio access networks." IEEE Transactions on Wireless Communications 14.11 (2015): 6164-6176.

[7] Corliano, A., and M. Hufschmid. Energieverbrauch der mobilen Kommunikation: Schlussbericht. Bundesamt für Energie, 2008.

[8] Geng, Zhiming, et al. "Performance analysis and comparison of GPP-based SDR systems." 2017 7th IEEE International Symposium on Microwave, Antenna, Propagation, and EMC Technologies (MAPE). IEEE, 2017.

[9] Rotem, Efraim, et al. "Power-management architecture of the intel microarchitecture code-named sandy bridge." Ieee micro 32.2 (2012): 20-27.

[10] Hackenberg, Daniel, et al. "Power measurement techniques on standard compute nodes: A quantitative comparison." 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, 2013.

[11] Mazouz, Abdelhafid, Benoît Pradelle, and William Jalby. "Statistical validation methodology of cpu power probes." Euro-Par 2014: Parallel Processing Workshops: Euro-Par 2014 International Workshops, Porto, Portugal, August 25-26, 2014, Revised Selected Papers, Part I 20. Springer International Publishing, 2014.

[12] 3GPP TS 36.213, “Evolved Universal Terrestrial Radio Access (EUTRA) Physical layer procedures (Release 8),” V9.0.1, (Dec. 2009).

[13] Kawser, Mohammad T., et al. "Downlink snr to cqi mapping for different multipleantenna techniques in lte." International journal of information and electronics engineering 2.5 (2012): 757.

[14] Budhdev, Nishant, Mun Choon Chan, and Tulika Mitra. "Pr 3: Power efficient and low latency baseband processing for lte femtocells." IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE, 2018.

[15] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer International Publishing, 2014

[16] Everingham, Mark, et al. "The pascal visual object classes challenge: A retrospective." International journal of computer vision 111 (2015): 98-136.

[17] 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.

Financiado por el Programa "Next Generation" de la Unión Europea en el marco del acuerdo n.º 2022/0004810