

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



*Trabajo Fin de Grado*

**ESTUDIO DEL CONSUMO ENERGÉTICO DE  
TÉCNICAS DE NETWORK CODING**  
(On the energy consumption of Network  
Coding techniques)

Para acceder al Título de

***Graduado en  
Ingeniería de Tecnologías de Telecomunicación***

Autor: David Cruz Trueba

Septiembre - 2019

# Agradecimientos

En primer lugar, me gustaría agradecer a Ramón Agüero, director de este trabajo, su continua ayuda durante los últimos meses, sin la cual no hubiese sido posible la realización del mismo.

También agradecer a Pablo Garrido sus indicaciones iniciales, y a Luis Francisco Diez por todas las dudas resueltas.

Por último, agradezco a mis amigos y a mi familia el apoyo recibido, no solo durante el desarrollo de este trabajo, sino en los últimos cuatro años en mi paso por la universidad.

# Resumen ejecutivo

En la actualidad, las redes de comunicaciones inalámbricas están en continuo crecimiento y, con la llegada de 5G, el número de dispositivos en estas redes va a aumentar considerablemente. Por ello, uno de los protocolos de transporte más utilizados hasta ahora, TCP, no parece una opción adecuada, ya que presenta problemas en redes propensas a errores. Las técnicas de *Network Coding* se han propuesto como una alternativa viable para este tipo de redes, y se ha demostrado que se pueden alcanzar velocidades aceptables. Sin embargo, no está claro el consumo energético adicional que estas técnicas introducen en una comunicación inalámbrica con respecto a otros protocolos. En este trabajo se estudia la energía adicional requerida por un dispositivo transmisor y otro receptor utilizando técnicas de *Network Coding*, combinadas con el protocolo UDP. Al final de este trabajo se recogen varios resultados relevantes, como el hecho de que se produzca un mayor consumo en el dispositivo receptor que en el transmisor, presentando tendencias de crecimiento diferentes, o variaciones considerables de dicho consumo con el uso de distintas características de codificación.

# Executive abstract

Nowadays, wireless networks are growing continuously, and the number of devices is expected to multiply, with the upcoming 5G technology. Thus, one of the most used transport protocols, TCP, seems not to be the best alternative to use over these error prone networks, where it has shown a bad behaviour. Network Coding techniques have been proposed as a possible alternative over these scenarios, and it has already been proved that it can work correctly. However, there are very few studies about the increase of the energy consumption these techniques would introduce. In this document, the additional energy consumption using Network Coding techniques, combined with UDP is studied. It shows that receiving devices consume more energy than transmitting devices, as well as a different growth trend in both scenarios. Finally, the energy consumption using different configurations of these techniques is also analyzed.

# Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación . . . . .	1
1.2	Objetivos . . . . .	2
1.3	Estructura . . . . .	3
<b>2</b>	<b>Antecedentes teóricos</b>	<b>4</b>
2.1	Técnicas de Network Coding . . . . .	4
2.2	Random Linear Network Coding . . . . .	8
2.3	Técnicas aplicadas a este TFG . . . . .	9
<b>3</b>	<b>Diseño e implementación</b>	<b>11</b>
3.1	Hardware . . . . .	11
3.1.1	Multímetro . . . . .	13
3.1.2	Raspberry Pi . . . . .	14
3.1.3	Otros componentes . . . . .	15
3.2	Software . . . . .	16
3.2.1	Programas TX y RX . . . . .	18
3.2.2	Matlab . . . . .	20
<b>4</b>	<b>Medidas y resultados</b>	<b>22</b>
4.1	Formato y características generales . . . . .	22
4.2	Consumo energético total en transmisión y recepción . . . . .	25
4.3	Consumo energético introducido por técnicas de Network Coding . . . . .	28
<b>5</b>	<b>Conclusiones</b>	<b>35</b>
5.1	Conclusiones . . . . .	35
5.2	Líneas futuras . . . . .	36
<b>A</b>	<b>Función visualizar</b>	<b>37</b>
<b>B</b>	<b>Función media</b>	<b>38</b>

# Lista de figuras

2.1	Grafo con codificación en nodos intermedios . . . . .	5
2.2	Pila de protocolos con y sin NC . . . . .	9
3.1	Esquema de Comunicación Inalámbrica . . . . .	12
3.2	Esquema de configuración hardware . . . . .	12
3.3	Multímetro Digital, modelo 34465 A de Keysight Technologies	13
3.4	Plataforma de medidas real . . . . .	16
4.1	Intensidad consumida por una Raspberry Pi en una comuni- cación inalámbrica. (a) Transmisor y (b) Receptor . . . . .	23
4.2	Intensidad media consumida por una Raspberry Pi en trans- misión . . . . .	26
4.3	Intensidad media consumida por una Raspberry Pi en recepción	28
4.4	Intensidad media introducida por las técnicas de NC en trans- misión . . . . .	30
4.5	Intensidad media introducida por las técnicas de NC en recepción	32
4.6	Comparativa de la intensidad requerida en transmisión y re- cepción para diferentes tamaños de campo finito. (a) $q = 1$ , (b) $q = 4$ , (c) $q = 8$ . . . . .	33
4.7	Intensidad (A) requerida en cada instante de tiempo (s) para diferentes tamaños de campo finito en transmisión. (a) $q = 1$ , (b) $q = 4$ , (c) $q = 8$ . . . . .	33

# Lista de tablas

3.1	Formato de almacenamiento del multímetro . . . . .	14
4.1	Intensidad media consumida por una Raspberry Pi en una transmisión inalámbrica (mA) . . . . .	25
4.2	Consumo Energético de una Raspberry Pi en una transmisión inalámbrica (J) . . . . .	26
4.3	Intensidad media consumida por una Raspberry Pi en una recepción inalámbrica (mA) . . . . .	27
4.4	Consumo Energético de una Raspberry Pi en una recepción inalámbrica (J) . . . . .	27
4.5	Intensidad adicional consumida por una Raspberry Pi en una transmisión inalámbrica debido a las técnicas de NC (mA) . . . . .	29
4.6	Consumo energético adicional de una Raspberry Pi en una transmisión inalámbrica debido a las técnicas de NC (J) . . . . .	29
4.7	Intensidad adicional consumida por una Raspberry Pi en una recepción inalámbrica debido a las técnicas de NC (mA) . . . . .	31
4.8	Consumo energético adicional de una Raspberry Pi en una recepción inalámbrica debido a las técnicas de NC (J) . . . . .	31

# Acrónimos

ACK	Acknowledgement.
GF	Galois Field.
IP	Internet Protocol.
LNC	Linear Network Coding.
NC	Network Coding.
RLNC	Random Linear Network Coding.
SSH	Secure Socket Shell.
TCP	Transmission Control Protocol.
TFG	Trabajo Fin de Grado.
UDP	User Datagram Protocol.

# Capítulo 1

## Introducción

En este capítulo se explica la motivación que impulsa a la realización de este Trabajo Fin de Grado (TFG), así como los objetivos planteados al inicio del mismo. Por último, se indica la estructura que sigue el documento.

### 1.1 Motivación

En la actualidad, los protocolos Transmission Control Protocol (TCP) y User Datagram Protocol (UDP) son las alternativas más utilizadas a nivel de transporte en Internet. El primero es un protocolo orientado a la conexión, que ofrece control de flujo y de errores, garantizando una recepción correcta y ordenada de todos los paquetes en el nivel de aplicación. Sin embargo, TCP no es un protocolo válido para todas las aplicaciones ya que, el consumo de recursos y los retardos temporales pueden limitar su aplicabilidad. Además, el número de redes inalámbricas es mucho mayor que en los orígenes de TCP, siendo estas un escenario poco favorable para el uso de dicho protocolo, debido a que son propensas a tener errores.

En su lugar, en aplicaciones para las cuales tanto el consumo de recursos como los retardos temporales son factores críticos, o en redes propensas a errores, es común utilizar UDP como protocolo de transporte. UDP es un protocolo no orientado a la conexión, que simplemente envía datagramas desde/hacia el nivel de aplicación, sin proporcionar ninguno de los mecanismos mencionados en TCP, pero mejorando tanto el consumo como los retardos temporales, además de no ser tan perjudicado en redes con tendencia a

errores.

Las técnicas de Network Coding (NC) se plantean como una posible solución intermedia, donde los problemas encontrados con TCP respecto a la congestión de las redes o los retardos temporales podrían minimizarse, garantizando además, a diferencia de UDP, una recepción ordenada por parte del nivel superior. Para ello, una posible solución es la combinación del protocolo UDP con las técnicas de NC.

Sin embargo, no está claro si esta combinación, debido a la inclusión de NC, será beneficiosa en cuanto a consumo energético se refiere. En este TFG se comprueba la variación de consumo energético de una transmisión inalámbrica, utilizando el protocolo UDP, combinado con diferentes técnicas de NC.

## 1.2 Objetivos

Las técnicas de NC han sido propuestas en diversas ocasiones como una solución a los problemas planteados en diferentes escenarios, como redes inalámbricas o sistemas con requisitos temporales estrictos. Sin embargo, algo que preocupa respecto a esta solución es su viabilidad, si se tiene en cuenta el consumo energético adicional que introducen.

Como se menciona en la sección anterior, este TFG estudia la variación de energía consumida por un transmisor a la hora de comunicarse con un receptor en una red inalámbrica utilizando técnicas de NC. Para ello, se compara el consumo energético de dos modos de transmisión diferentes. El primero utiliza el protocolo UDP en la capa de transporte de una manera tradicional mientras que el segundo, combina este protocolo con diferentes técnicas de Network Coding.

En conclusión, este trabajo valora en qué medida la utilización de las técnicas de NC incrementan el consumo energético de las comunicaciones inalámbricas, comprobando su viabilidad, y determinando la configuración óptima de estas.

## 1.3 Estructura

Este documento consta de cinco capítulos en los que se incluye la siguiente información:

- Este primer capítulo presenta la motivación de este trabajo, y los objetivos que se pretenden alcanzar al final del mismo.
- El segundo capítulo expone los fundamentos teóricos necesarios para la comprensión del propio trabajo, explicando brevemente las técnicas de Network Coding, y sus características principales.
- El tercer capítulo describe el diseño y la implementación de la plataforma de medidas, incluyendo una explicación de todo el material necesario, tanto hardware como software, para su correcto funcionamiento.
- El cuarto capítulo recoge los resultados obtenidos a lo largo del desarrollo de este trabajo, comparando el consumo energético medido en la plataforma implementada, utilizando las diferentes técnicas de transmisión descritas en los Capítulos 2 y 3.
- El quinto y último capítulo presenta para las conclusiones finales obtenidas a partir de todo el proceso realizado durante el desarrollo del TFG.

# Capítulo 2

## Antecedentes teóricos

Durante todo este trabajo se hace referencia en numerosas ocasiones a las técnicas de NC, siendo el objetivo final del mismo calcular el aumento del consumo energético que estas introducen. Este capítulo contiene una breve explicación de las técnicas de NC, y cuáles son las características más relevantes para el desarrollo de este TFG ya que, en capítulos posteriores se habla de las mismas en diversas ocasiones.

### 2.1 Técnicas de Network Coding

Como se ha mencionado en la introducción, las técnicas de NC han sido propuestas como una solución intermedia entre los protocolos TCP y UDP, mejorando los problemas de congestión y retardos que sufre TCP, generalmente, en redes inalámbricas. Pero, ¿qué son las técnicas de Network Coding? En la tesis doctoral *Opportunistic Network Coding over Wireless Networks* [1] se recoge una explicación muy detallada, que se resume a continuación, centrándose en los puntos más relevantes para el desarrollo de este trabajo.

En sus orígenes, el término NC se acuñó en [2], refiriéndose a la posibilidad de introducir codificación en los nodos intermedios de una red, dejando de un lado el trato de la información como un “fluido” que solo puede ser transmitido, como era lo habitual hasta entonces. Para facilitar su comprensión se utiliza una representación gráfica del problema y su solución.

En la Figura 2.1 aparecen cuatro grafos en los que un nodo fuente S genera

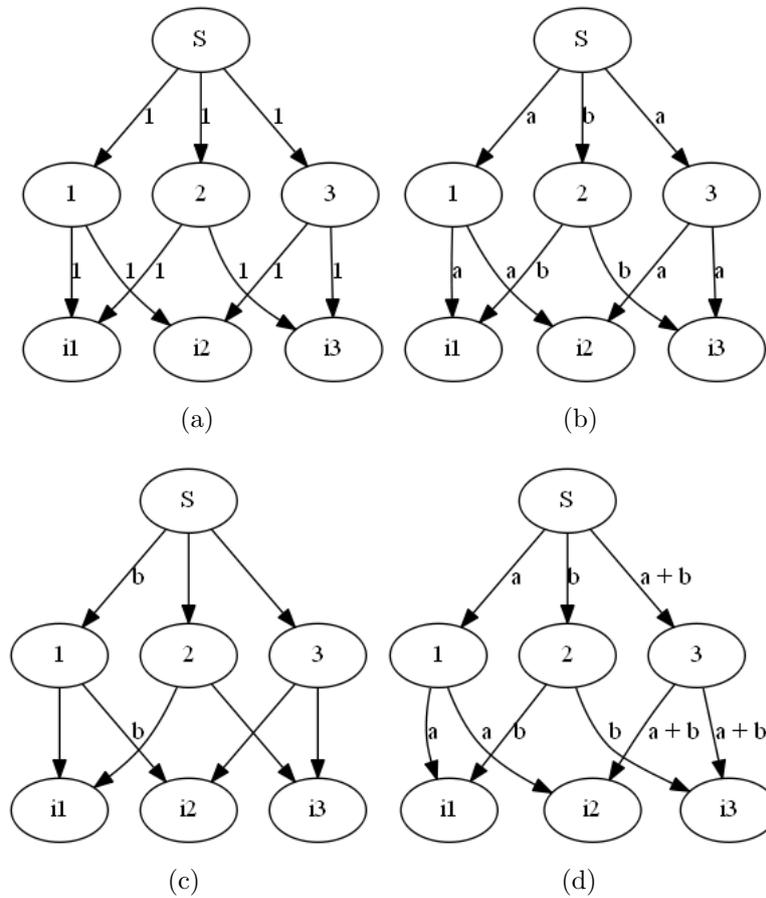


Figura 2.1: Grafo con codificación en nodos intermedios

y envía información a los nodos 1, 2 y 3, y estos, a su vez, se la entregan a los nodos  $i_1$ ,  $i_2$  y  $i_3$ . El objetivo es enviar a los tres nodos  $i$  dos paquetes ( $a$  y  $b$ ) de longitud genérica unidad, transmitidos por  $S$  y que estos puedan posteriormente enviarlos a nuevos destinatarios. Para ello, es necesario que cada uno de los tres nodos  $i$  posean los dos paquetes individuales.

En la Figura 2.1a se representa la capacidad de cada enlace entre dos nodos,  $\text{Nodo } S \rightarrow \text{Nodo } 1$ , como una unidad. Si se tratara la información como un fluido, es decir, los nodos intermedios en una red simplemente reciben la información y no la procesan, para enviar los dos paquetes a los tres nodos  $i$ ,  $S$  debería mandar los paquetes como se muestra en la Figura 2.1b, y posteriormente utilizar de nuevo el enlace  $S \rightarrow 1$  o  $S \rightarrow 3$  para que el nodo  $i_2$  pudiese recibir el paquete  $b$ , como se muestra en la Figura 2.1c. Sin embargo, si se introduce NC y todos los nodos (en este caso con los nodos  $i$  sería sufi-

cientemente) pueden codificar y decodificar la información, transmitiendo por uno de los enlaces una combinación de ambos paquetes, como se muestra en la Figura 2.1d, el envío adicional mostrado en la Figura 2.1c se podría evitar.

Así, el nodo  $i1$  tendría el paquete  $a$  y  $b$  mientras que los nodos  $i2$  e  $i3$  tendrían el paquete  $a$  ó  $b$ , y una combinación de los paquetes  $a$  y  $b$ . Sin embargo, gracias al uso de NC, los nodos  $i2$  e  $i3$  serían capaces de recuperar el paquete individual que les falta, a partir del otro y la combinación de ambos. De esta manera, el nodo  $S$  se evitaría una transmisión innecesaria y todos los nodos  $i$  recibirían los dos paquetes  $a$  y  $b$ .

Posteriormente, en los artículos [3] y [4] se habla de Linear Network Coding (LNC), donde se introduce el concepto de cuerpo finito Galois Field (GF), siendo la información a enviar elementos pertenecientes a este. El cuerpo finito  $GF(2^q)$  está definido por cuatro operaciones: suma, resta, multiplicación y división, permitiendo a los nodos de una red que dispongan de LNC codificar la información utilizando cualquiera de estas operaciones. Para ello, se sigue la siguiente ecuación:

$$p'_i = \vec{c}_i \cdot \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,m} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,m} \\ \dots & \dots & \dots & \dots \\ p_{k,1} & p_{k,2} & \cdots & p_{k,m} \end{bmatrix} \quad (2.1)$$

En la ecuación 2.1,  $p'_i = [p'_1, p'_2, \dots, p'_m]$  se corresponde con el paquete finalmente codificado, que surge de la combinación lineal de cualquiera de los paquetes nativos que forman la matriz inicial. Dicha combinación depende del vector  $c_i = [c_1, c_2, \dots, c_k]$ , que contiene los coeficientes de codificación. Todos los paquetes nativos en la matriz están compuestos por el mismo número de elementos, comenzando por  $[p_{1,1}, p_{1,2}, \dots, p_{1,m}]$  y terminando por  $[p_{k,1}, p_{k,2}, \dots, p_{k,m}]$ .

Para tener una visión más general y básica de qué son las técnicas de NC se plantea la situación del envío de cierta información, desde un origen a un destino sobre un único enlace inalámbrico, sin nodos intermedios. No se trata por tanto de una aplicación real, ya que NC se plantea como una solución en redes con muchos usuarios, como se ha mencionado en el Capítulo 1. Sin embargo, para la finalidad de este trabajo, que es medir el consumo energético que causan estas técnicas, es perfectamente válido, y por tanto, la forma elegida para el desarrollo del TFG.

En general, en la transmisión de dicha comunicación se podrán configurar diferentes parámetros, como el número de generaciones, el número de paquetes por generación, el tamaño de los símbolos,  $k$ , o el número de elementos,  $2^q$ , en GF.

- Símbolos: para transmitir la información mencionada, en primer lugar se generan los símbolos, siendo estos fragmentos de dicha información, divididos en partes iguales. A la hora de configurar las técnicas de NC se podrán elegir el número de símbolos, y el tamaño de los mismos, siempre y cuando el número de símbolos por el tamaño de estos coincida con el total de la información a enviar.
- Paquetes: los símbolos generados anteriormente se corresponden con la información nativa que se desea enviar. Para ello, como se indica en la ecuación 2.1, se crean paquetes compuestos por combinaciones lineales de dichos símbolos. La información que viaja en estos paquetes es del mismo tamaño que los símbolos debido a que, como se ha explicado anteriormente, el cuerpo finito que los contiene está definido por operaciones lineales. Sin embargo, se añade una pequeña cabecera adicional, con información sobre los parámetros de configuración elegidos en la codificación, y que es necesaria en el receptor.
- Generaciones: para enviar los paquetes al destinatario, el transmisor los agrupa en generaciones, esperando la recepción de un Acknowledgement (ACK) antes de continuar con la siguiente generación. Esto se debe a que el transmisor no sabe cuántos paquetes va a necesitar el receptor para recuperar la información final. Como los paquetes están formados por combinaciones lineales de los símbolos, si el receptor recibe paquetes que son linealmente dependientes entre sí, los descarta y espera hasta recibir alguno válido (linealmente independiente). Por ello, el número de paquetes enviados y recibidos siempre será igual o mayor que el número de símbolos a transmitir y recuperar.
- $GF(2^q)$ : el cuerpo finito seleccionado en las técnicas de NC fija el número de opciones que los coeficientes de codificación mencionados en la fórmula 2.1 pueden tomar. El cuerpo finito puede tener distintos tamaños, dependiendo de las características que se deseen para las técnicas de NC, siendo el menor  $GF(2^1)$ , donde dichos valores podrán ser 0 o 1.  
Se puede ver una idea más práctica en la ecuación 2.2 y su correspondiente explicación.

## 2.2 Random Linear Network Coding

Las técnicas de Random Linear Network Coding (RLNC) fueron propuestas en [5] y completadas posteriormente en [6], donde se presenta un enfoque de RLNC para la transmisión de información en redes multisalto.

En los citados trabajos se expone que, si el campo finito elegido para la matriz de codificación es suficientemente grande, la elección de RLNC consigue la solución óptima, del mismo modo que lo hacen otras técnicas.

La diferencia entre RLNC y otras técnicas de NC recae en la elección de la matriz de codificación  $C$ , mencionada en la ecuación 2.1. En otras técnicas se emplea una lógica específica para la creación de esta matriz, buscando la solución óptima para que, a la hora de codificar, la probabilidad de enviar un paquete linealmente dependiente sea la mínima. En dichos trabajos se demuestra que, si  $\text{GF}(2^q)$  es lo suficientemente grande, se pueden elegir los coeficientes de esta matriz de forma aleatoria, obteniendo de igual manera la solución óptima en la codificación.

$$\begin{bmatrix} p'_1 \\ p'_2 \\ p'_3 \\ \dots \\ p'_m \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,3} & \dots & c_{1,k} \\ c_{2,1} & c_{2,2} & c_{2,3} & \dots & c_{2,k} \\ c_{3,1} & c_{3,2} & c_{3,3} & \dots & c_{3,k} \\ \dots & \dots & \dots & \dots & \dots \\ c_{m,1} & c_{m,2} & c_{m,23} & \dots & c_{m,k} \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \dots \\ p_k \end{bmatrix} \quad (2.2)$$

En la ecuación 2.2 se muestra la matriz  $C$  compuesta por todos los coeficientes de codificación que, en las técnicas de RLNC se obtienen de manera aleatoria. Es importante que el campo finito elegido sea lo suficientemente grande ya que, de no ser así, la probabilidad de obtener paquetes linealmente dependientes sería mucho mayor.

Es lógico pensar que si dichos coeficientes pertenecen al cuerpo finito binario  $\text{GF}(2^1)$ , siendo este el tamaño más pequeño en el que los únicos valores posibles son 0 y 1, es más probable que surjan paquetes linealmente dependientes que si, por ejemplo, dichos coeficientes pertenecen a  $\text{GF}(2^8)$ , donde cada uno podrá tomar 1 valor entre 256 opciones.

Además, se ha demostrado que, a la hora de decodificar la información, los cuerpos finitos de mayor tamaño tienen un mejor comportamiento, necesitando en media menos paquetes codificados en el receptor para completar la

decodificación. Esto no ocurre solo cuando se aplican las técnicas de RLNC, si no con cualquier técnica de NC, y está directamente relacionado con la cantidad de paquetes linealmente dependientes que se generan en los diferentes cuerpos finitos.

Sin embargo, como se ha mencionado anteriormente, el objetivo de este TFG es medir el consumo energético debido a estas técnicas y parece razonable que, cuanto mayor sea el cuerpo finito elegido, mayor será el consumo energético en transmisión, y sobre todo, en recepción.

## 2.3 Técnicas aplicadas a este TFG

Como ya se ha mencionado, durante el desarrollo de este trabajo se pretende conocer el consumo adicional que introducen las técnicas de NC en comunicaciones inalámbricas. Para realizar todas las pruebas y obtener los resultados que se describen en los capítulos posteriores se han utilizado las técnicas de RLNC descritas en la sección anterior ya que los programas explicados en el Capítulo 3 y utilizados durante este TFG hacen uso de las mismas. Por ello, todos los resultados y conclusiones finales se corresponden a consumos debidos únicamente a RLNC, y no a otras técnicas que puedan aparecer en diferentes documentos referenciados durante este trabajo.

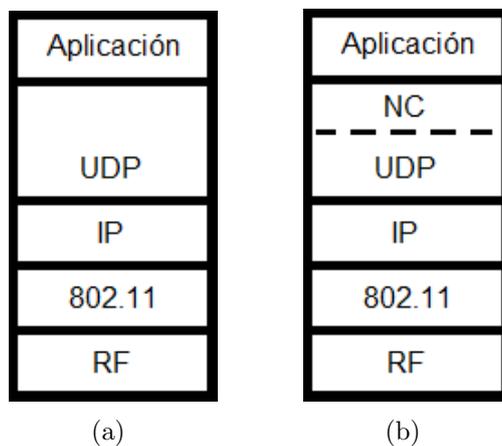


Figura 2.2: Pila de protocolos con y sin NC

Otro punto a tener en cuenta es el lugar en donde las técnicas de NC se introducen dentro de la pila de protocolos.

Diferentes estudios plantean su situación en distintas capas de la pila de protocolos, abarcando desde la capa de enlace hasta el inicio de la capa de aplicación. Entre todas las opciones, en este trabajo se ha decidido combinar dichas técnicas con el protocolo UDP en la capa de transporte como se muestra en la Figura 2.2. En ella se puede observar el modelo tradicional que se utiliza para medir el consumo energético de una transmisión inalámbrica convencional y poder compararlo con el consumo medido en una transmisión inalámbrica según la arquitectura de la Figura 2.2b, donde se introducen las técnicas de NC. De esta manera, calculando la diferencia de energía consumida entre ambas transmisiones se puede conocer el consumo adicional que introducen únicamente las técnicas de NC, que es el objetivo final de este trabajo.

Además del tipo de técnicas de NC utilizadas y el lugar que ocupan dentro de la pila de protocolos, es conveniente introducir sus distintos parámetros de configuración. Durante el desarrollo de este trabajo se van a variar principalmente el número de símbolos/paquetes por generación y el valor de  $GF(2^q)$ , con el fin de conocer el consumo energético utilizando diferentes configuraciones.

Teóricamente, aumentando el número de símbolos por generación también se aumenta el consumo energético, sobre todo en recepción, debido a que, para cada paquete recibido, el receptor deberá comprobar si este es linealmente dependiente con otros recibidos anteriormente. Como parece obvio, cuantos más paquetes se hayan recibido, más posibilidades existen de que estos sean dependientes de algún otro y, por lo tanto, los vuelva inservibles.

En cuanto a la variación de  $q$ , como ya se ha mencionado en la sección anterior, el aumento de este parámetro consigue un mayor abanico de posibilidades en la elección de los coeficientes de codificación de la matriz  $C$ . Es interesante tomar medidas con diferentes tamaños de  $GF$  ya que, aunque se ha demostrado que un mayor tamaño del cuerpo finito conlleva un mejor comportamiento a la hora de decodificar (puesto que es necesario un menor número de paquetes codificados en el receptor para obtener toda la información), este TFG se centra en el consumo energético que introducen las técnicas de NC en una comunicación inalámbrica, siendo este mayor cuanto más aumente  $q$ . Por ello, si se conocen tanto la probabilidad de decodificación dependiendo de  $GF(2^q)$ , como la variación del consumo introducida para distintos valores de  $q$ , se podrán tomar soluciones de compromiso dependiendo de cada situación.

# Capítulo 3

## Diseño e implementación

A lo largo de este capítulo se expone una descripción detallada de todo el material necesario para el diseño y despliegue de la plataforma de medidas utilizada en este trabajo. Está dividido en dos grandes secciones: la primera explica el funcionamiento y las características principales de todos los componentes físicos utilizados. Por su parte, la segunda se centra en los programas necesarios.

### 3.1 Hardware

Como se explica en capítulos anteriores, el objetivo final de este TFG es conocer el aumento del consumo energético de un dispositivo, en este caso una Raspberry Pi 3 B+, cuando utiliza técnicas de NC para la transmisión y recepción de datos a través de un medio inalámbrico, en comparación con la misma comunicación inalámbrica utilizando el protocolo UDP de manera tradicional.

En la Figura 3.1 se puede observar un esquema de los elementos que intervienen en la comunicación.

En este diagrama aparecen dos Raspberry Pi 3 B+, que son las encargadas de comunicarse entre sí a través de su interfaz radio, utilizando primero el protocolo UDP de manera convencional y después, combinándolo con técnicas de NC.

Para poder medir la energía consumida por las Raspberry Pi, se utiliza



Figura 3.1: Esquema de Comunicación Inalámbrica

un Multímetro digital en modo amperímetro, que conectado en serie a una de estas, proporciona la intensidad que consume.

En el esquema de la Figura 3.1 aparecen únicamente dos Raspberry Pi y un multímetro digital funcionando como amperímetro. Sin embargo, para controlar los dispositivos transmisor y receptor y almacenar los datos obtenidos por el multímetro, es necesario incorporar un ordenador a la plataforma de medidas.

Como se puede apreciar en la Figura 3.2, el ordenador se conecta a ambas Raspberry Pi mediante cables ethernet, que pasan por un switch intermedio para poder trabajar en la misma red local. Por otro lado, para volcar los datos recogidos por el multímetro, este se conecta al ordenador mediante un cable USB proporcionado por el fabricante del mismo.

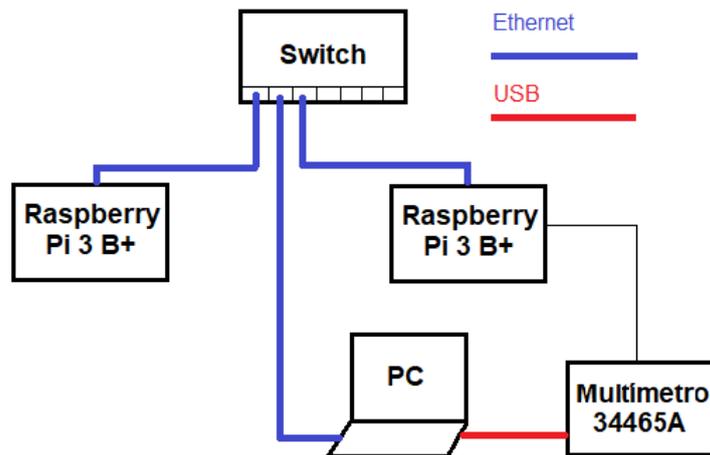


Figura 3.2: Esquema de configuración hardware

### 3.1.1 Multímetro

El multímetro digital es un componente imprescindible en la plataforma de medidas a desarrollar para la realización de este TFG, ya que es el encargado de tomar las medidas de consumo energético que se presentan en el Capítulo 4.

Sin embargo, no todos los multímetros son válidos para desempeñar esta tarea. Los tiempos que manejan los programas descritos posteriormente son del orden de los micro segundos, haciendo inviable el uso de un multímetro convencional, cuya frecuencia máxima suele rondar los 500 Hz.

Por ello, el multímetro digital elegido para incluir en la plataforma de medidas ha sido el Modelo 34465 A del fabricante Keysight Technologies (ver Figura 3.3) cuya frecuencia máxima alcanza los 300 kHz.



Figura 3.3: Multímetro Digital, modelo 34465 A de Keysight Technologies

Este dispositivo cuenta con múltiples opciones. Sin embargo, durante el desarrollo de este trabajo se han utilizado principalmente las mencionadas a continuación:

- La opción DCI, que permite medir corriente continua. Para utilizar esta opción es necesario pulsar las teclas ‘Shift’ + ‘DCV’.
- El modo Digitize, que permite almacenar los resultados en una hoja de cálculo con el formato que se muestra en la Tabla 3.1.

Para acceder a este modo es necesario pulsar la tecla ‘Acquire’ y seleccionar la opción ‘Digitize’, que aparece en la parte inferior izquierda

del panel. Dentro de esta opción se pueden ajustar parámetros como el intervalo entre muestras, el tiempo de duración de la medida, o algunas características del disparo. Una vez configurados los parámetros deseados, basta con pulsar el botón ‘Run/Stop’ para iniciar la medida. Posteriormente, bien pulsando de nuevo este botón, o esperando a que termine automáticamente, se pueden guardar los datos recogidos pulsando ‘Save’, en la parte inferior derecha de la pantalla, obteniendo la hoja de cálculo previamente mencionada.

Tabla 3.1: Formato de almacenamiento del multímetro

Start Date: dd/mm/yyyy	Start Time: hh:mm:ss
Sample Interval:	(segundos)
Reading#	Reading
1	a
2	b
3	c
...	...
n	m

Esta opciones y todas las demás funcionalidades del multímetro están explicadas detalladamente en el manual del mismo, que se puede encontrar en la página principal del fabricante<sup>1</sup>.

### 3.1.2 Raspberry Pi

Como se ha explicado anteriormente, la plataforma de medidas cuenta con dos Raspberry Pi 3 modelo B+ que actúan, como transmisor y receptor. A diferencia de modelos anteriores, estos dispositivos cuentan con una tarjeta de red incorporada que soportan los estándares IEEE 802.11.b/g/n/ac, por lo que no es necesario incluir un dispositivo adicional para habilitar la comunicación inalámbrica entre ellas. Además, cuentan con una entrada RJ45 que permite, utilizando un cable ethernet, su conexión con un PC y, mediante Secure Socket Shell (SSH), controlar las operaciones a realizar.

<sup>1</sup><https://www.keysight.com/>

### 3.1.3 Otros componentes

Como se puede apreciar en la Figura 3.1, ambas Raspberry Pi están alimentadas por una fuente de alimentación continua de 5 voltios. Sin embargo, se ha tenido que hacer una pequeña modificación en una de las fuentes proporcionadas por el fabricante (transformador) debido a la opción elegida para la implementación de la plataforma. Para realizar las medidas del consumo energético necesarias en este TFG se han planteado dos opciones relevantes:

La primera opción utiliza el multímetro digital mencionado anteriormente como voltímetro, que conectado en paralelo con una pequeña resistencia al circuito formado por la fuente de alimentación y la Raspberry Pi, proporciona el voltaje empleado por el dispositivo en el momento deseado. A partir de este voltaje, calcular la energía consumida es una operación directa.

Sin embargo, la opción elegida ha sido la segunda. Con el multímetro en modo amperímetro, se conecta en serie al circuito mencionado anteriormente y así, se obtiene la intensidad consumida por la Raspberry Pi en cualquier momento. Una vez se conoce la intensidad, simplemente se multiplican los 5 voltios continuos que proporciona la fuente para obtener la potencia, y por último, se tiene en cuenta el tiempo empleado, para obtener la energía consumida por el dispositivo:

$$E = I * V * t \quad (3.1)$$

Por último, como se muestra en la imagen real de la plataforma de medidas, Figura 3.4, para conectar el multímetro en serie al circuito ha sido necesario realizar un corte en el cable de alimentación. Después, con ayuda de una tabla de prototipos, dos clemas y unos cables auxiliares se ha realizado la conexión final, permitiendo así medir la energía consumida por cualquier dispositivo alimentado con dicha fuente. En esta imagen aparecen todos los dispositivos mencionados, identificados de izquierda a derecha como Raspberry Pi transmisora, switch, ordenador, placa de prototipos, Raspberry Pi receptora y multímetro digital.



Figura 3.4: Plataforma de medidas real

## 3.2 Software

Si bien la parte hardware de la plataforma explicada hasta ahora es esencial, sin los programas adecuados, no tendría ninguna utilidad para conseguir el objetivo final de este TFG. En primer lugar, las Raspberry Pi 3 B+ no tienen ningún sistema operativo por defecto. Para poder trabajar con ellas, se ha optado por introducir el sistema operativo Raspbian, que se puede descargar de manera gratuita desde la página web del fabricante<sup>2</sup>.

Por otro lado, como se ha mencionado en la sección anterior, las Raspberry Pi están conectadas al ordenador mediante cables ethernet con la ayuda de un switch. Para controlarlas con el PC se ha utilizado el cliente SSH Putty, también gratuito y de fácil acceso a través de su página web<sup>3</sup>. Para facilitar el continuo uso de la plataforma, se ha optado por configurar direcciones Internet Protocol (IP) estáticas y en la misma subred, en los 3 dispositivos. De esta manera, con introducir la dirección IP de las Raspberry Pi en el cliente

---

<sup>2</sup><http://www.raspberrypi.org>

<sup>3</sup><http://www.putty.org>

Putty descargado e instalado en el ordenador, se iniciará la conexión SSH automáticamente.

Para configurar las direcciones IP estáticas en las Raspberry Pi, es necesario editar el archivo `/etc/dhcpd.conf`, con las siguientes líneas:

```
interface eth0
static ip_address=192.168.0.10/24
```

Una vez guardados los cambios, es necesario reiniciar el dispositivo para aplicarlos. Con el comando `ifconfig` se puede comprobar que la nueva dirección IP estática se ha asignado con éxito.

En este punto, las Raspberry Pi ya se pueden controlar desde el ordenador de una manera sencilla y rápida. Ahora es necesario que haya comunicación inalámbrica entre ambas Raspberry Pi, para posteriormente enviar los datos codificados con NC y medir su consumo energético.

Para ello, se han barajado las dos opciones explicadas en [7]: la configuración en modo Ad-Hoc, y el modo infraestructura, optando finalmente por la segunda opción. En la configuración en modo infraestructura, una de las Raspberry Pi actúa como punto de acceso y la otra se conectará a esta, estableciendo así la conexión a nivel de enlace 802.11. Al igual que con la interfaz ethernet, se han asignado IP estáticas a la interfaz wlan de ambos dispositivos.

Para configurar la primera como un punto de acceso son necesarias una serie de modificaciones en sus archivos de configuración:

En primer lugar, es necesario instalar el paquete `hostapd` para poder generar el punto de acceso. Una vez instalado se necesita configurar; editando el archivo `/etc/hostapd/hostapd.conf` con la siguiente información:

```
interface=wlan0
ssid=PiDavid
hw_mode=g
channel=6
ieee80211n=1
beacon_int=100
macaddr_acl=0
```

```
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_passphrase=raspberry
```

Por último, editando el archivo `/etc/default/hostapd` con la siguiente línea, se consigue iniciar el punto de acceso cada vez que se encienda la Raspberry Pi.

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

En este punto, y una vez reiniciado el primer dispositivo para que se apliquen los cambios realizados en su configuración, la segunda Raspberry Pi solo necesita conectarse a la primera para finalizar la conexión. Con el comando `ping 172.24.1.2 -I wlan0` se puede comprobar que la configuración funciona correctamente, y ambos dispositivos se pueden comunicar a través de su interfaz inalámbrica.

La conexión entre ambos dispositivos ya está establecida. El último paso consiste en cargar los programas transmisor y receptor en las Raspberry Pi para poder empezar a tomar medidas de consumo energético en comunicaciones con y sin NC.

### 3.2.1 Programas TX y RX

Durante el desarrollo de este trabajo se han utilizado cuatro programas diferentes, dos para transmitir y dos para recibir. Entre los dos programas transmisores, al igual que entre los dos en recepción, la única diferencia consiste en la inclusión de técnicas de NC en uno de ellos. Es decir, se cuenta con un transmisor que utiliza el protocolo UDP de manera convencional, por encima de IP, y otro, también transmisor, que es prácticamente igual, pero incluyendo las técnicas de codificación de NC combinadas con UDP. Así, la diferencia de consumo en la transmisión en un mismo dispositivo utilizando estos dos programas se debe únicamente a las técnicas de NC. De esta manera, es sencillo calcular la diferencia entre ambos casos, obteniendo directamente el consumo energético debido a las técnicas de NC, objetivo final de este TFG. En el caso del receptor se sigue exactamente el mismo

proceso, utilizando los dos programas receptores, con y sin la inclusión de técnicas de NC, calculando la diferencia de consumo entre ambos.

El funcionamiento de estos programas es el que se describe seguidamente:

El programa transmisor genera una serie de datos aleatorios del tamaño que se desee, y posteriormente los envía en bloques de bytes de longitud a elegir. La diferencia entre ambos programas transmisores se encuentra en este punto. Uno de ellos envía los datos utilizando el protocolo UDP de manera convencional, mientras que el otro los codifica utilizando las técnicas de NC explicadas en el Capítulo 2 de este documento.

Una vez los datos llegan al receptor, este enviará un paquete ACK cuando reciba el número de símbolos por generación establecido en la configuración previa. Si el programa no está utilizando las técnicas de NC, y suponiendo que ningún paquete se pierde, el transmisor enviará tantos paquetes en cada generación como número de símbolos se haya establecido, y el receptor enviará el ACK correspondiente a la recepción del último de ellos en cada generación. Es decir, si se ha configurado el envío del ACK a la recepción de  $n$  símbolos, se enviarán y recibirán  $n$  paquetes por generación.

Sin embargo, si se utilizan las técnicas de NC esto cambia. El número de paquetes enviados y recibidos puede ser superior al número de símbolos inicialmente configurados. Este incremento se debe a que es posible, dependiendo de los parámetros de configuración seleccionados, los paquetes recibidos en el receptor no sean todos linealmente independientes. De esta manera, si se ha configurado el envío del ACK a la recepción de  $n$  símbolos, se enviarán y recibirán  $n$  más  $m$  paquetes por generación, correspondiéndose estos últimos a símbolos linealmente dependientes.

Al inicio de los programas transmisor y receptor se pueden configurar algunos de los parámetros explicados en el Capítulo 2, con el fin de obtener resultados para distintas configuraciones. Estos parámetros son:  $GF(2^q)$ , tamaño de los símbolos, número de generaciones y número de símbolos por generación. Además, para facilitar la identificación visual de cada símbolo, se introduce un tiempo de espera antes del envío de los mismos.

Cabe mencionar que estos programas no han sido creados desde cero para la realización de este TFG. En su lugar, se han reutilizado los programas creados durante el desarrollo de la tesis doctoral *Opportunistic Network Coding over Wireless Networks* [1], donde se puede encontrar más información acerca de los mismos.

Así, al inicio de este trabajo se contaba con dos programas, uno transmisor y otro receptor, que incluían las técnicas de NC por defecto. Durante el desarrollo del mismo, y para conseguir que la medida en el consumo energético sea lo más precisa posible, se han modificado ambos programas eliminando las técnicas de NC asociadas a los mismos e introduciendo un tiempo de espera entre transmisiones.

En este punto es posible establecer la comunicación entre ambas Raspberry Pi utilizando o no las técnicas de NC, midiendo el consumo energético durante el proceso. A continuación, el último paso consiste en almacenar los datos medidos por el multímetro digital y procesarlos para obtener los resultados deseados.

Para ello se ha utilizado el software Matlab cuya página web oficial<sup>4</sup> proporciona mucha documentación acerca de su funcionamiento, utilizada parte de ella para la realización de este trabajo.

### 3.2.2 Matlab

Como se ha comentado a lo largo de este capítulo, el programa que se utiliza para procesar, analizar y representar los datos de consumo obtenidos con la plataforma previamente explicada es Matlab. Este software permite recoger los datos almacenados por el multímetro digital directamente de la hoja de cálculo que, como se ha explicado anteriormente, genera en cada experimento.

El proceso llevado a cabo en Matlab ha sido el siguiente:

En primer lugar, se recogen los datos de la hoja de cálculo correspondiente, y se almacenan en matrices dentro del propio programa, para facilitar su utilización. Como se muestra en la Tabla 3.1, el formato de las medidas de intensidad almacenadas en la hoja de cálculo se corresponde con una matriz de dos columnas, y tantas filas como medidas se tomen en un intervalo de tiempo definido. Por ello, el siguiente paso consiste en, mediante la función ‘visualizar’ incluida en el Anexo A, representar estos valores en una gráfica, correspondiéndose el eje de abscisas con el tiempo del experimento mientras que el eje de ordenadas representa el valor de la intensidad relativo a cada muestra. Este paso es opcional, sin embargo, es bastante útil para poder visualizar lo que se está transmitiendo realmente. Además, se puede recu-

---

<sup>4</sup><https://es.mathworks.com/products/matlab.html>

rrir a ella si en algún momento los resultados obtenidos durante el resto del proceso resultan inesperados, pudiendo determinar el origen de los mismos.

Como se ha mencionado anteriormente, las técnicas de NC cuentan con una serie de parámetros configurables, que las dotan de unas u otras características. Para comprobar cómo afectan estas al consumo energético, se han tomado medidas utilizando diferentes configuraciones. Además, con el fin de obtener resultados lo más precisos posibles, se han realizado varias medidas para cada configuración y, posteriormente, se ha obtenido la media de cada grupo, con la función ‘Media’ incluida en el Anexo B.

Esta función calcula la esperanza matemática de la intensidad consumida por la Raspberry Pi durante el tiempo que dure cada experimento y, posteriormente, calcula la media de esos resultados para un grupo de medidas con las mismas configuraciones.

En este punto se conoce la intensidad media consumida por la Raspberry Pi durante la transmisión/recepción de datos, utilizando los cuatro programas explicados en la sección anterior, es decir, el consumo en las transmisiones y recepciones tanto sin las técnicas de NC activas como con ellas activas, con las diferentes configuraciones elegidas.

El siguiente paso consiste en calcular el consumo energético debido únicamente a las técnicas de NC. Para ello, se calcula la diferencia de intensidad medida entre los consumos, utilizando NC y no. Como se ha explicado anteriormente, la única diferencia entre los programas utilizados es que una pareja incluye las técnicas de NC y la otra no, por lo que, la diferencia de consumo entre ambos se debe a este factor.

Por último, una vez que se conocen los consumos energéticos que introducen las técnicas de NC con sus variantes en la configuración, se representan los datos en gráficas para facilitar su lectura e interpretación y así, conseguir encontrar un conjunto de conclusiones.

# Capítulo 4

## Medidas y resultados

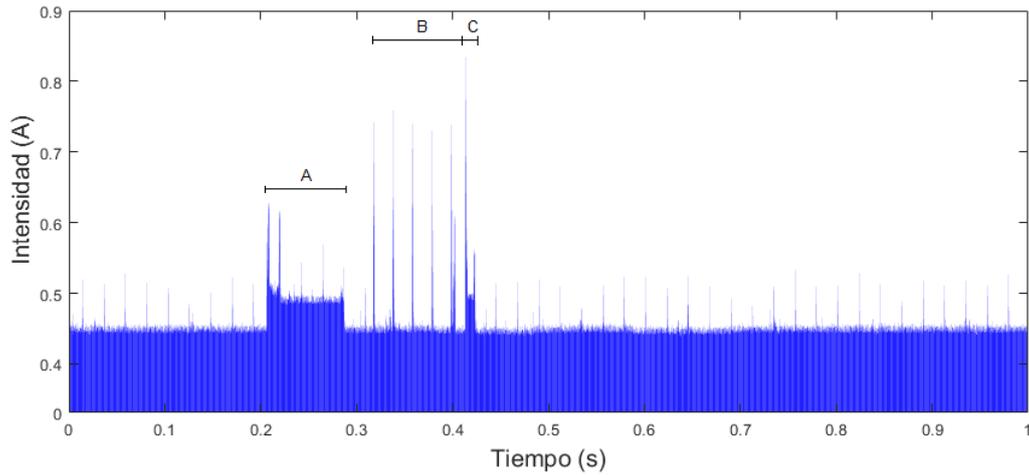
Este capítulo expone todos los resultados obtenidos con la plataforma de medidas descrita en el capítulo anterior, comparando la energía consumida por una Raspberry Pi en una comunicación inalámbrica, utilizando como protocolo de transporte UDP, y este mismo combinado con las técnicas de NC explicadas en el Capítulo 2, y algunas de sus posibles configuraciones.

### 4.1 Formato y características generales

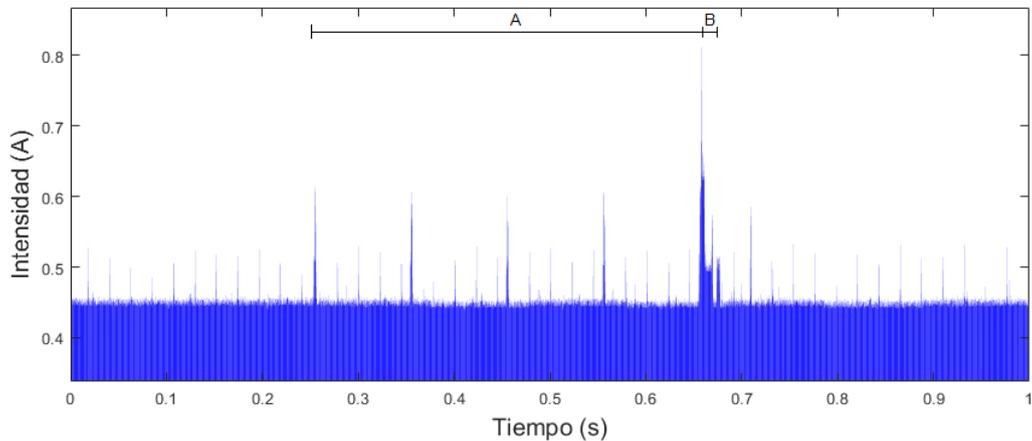
Antes de analizar los datos capturados por el multímetro, es interesante ver cómo se reparte en el tiempo la energía consumida en una comunicación inalámbrica en general. En la Figura 4.1a se muestra la intensidad consumida por una Raspberry Pi transmitiendo a través del medio inalámbrico utilizando uno de los programas descritos en el capítulo anterior. Como se puede observar, la intensidad que consume el dispositivo en reposo ronda los 450 mA, aumentando considerablemente durante los intervalos A, B y C.

En primer lugar, el intervalo A se corresponde con la generación de la información a enviar, además de la creación de los paquetes formados por combinaciones lineales de los símbolos cuando se utilizan las técnicas de NC. Por ello, en las medidas expuestas posteriormente, durante el intervalo A se apreciará un mayor consumo cuanto mayor sea el tamaño del campo finito, debido a que, como se ha explicado en el Capítulo 2, un mayor tamaño de GF aumenta la complejidad de la matriz de codificación.

El intervalo B recoge la transmisión de todos los paquetes. En el caso de la imagen, el transmisor envía cinco paquetes al receptor, llegando a una intensidad máxima de, aproximadamente, 750 mA en cada transmisión. El intervalo C se corresponde con la recepción del ACK enviado por el receptor una vez que este recibe toda la información necesaria.



(a)



(b)

Figura 4.1: Intensidad consumida por una Raspberry Pi en una comunicación inalámbrica. (a) Transmisor y (b) Receptor

Por otro lado, la Figura 4.1b representa la intensidad consumida por una Raspberry Pi de una comunicación similar a la anterior. Al igual que en transmisión, la intensidad consumida por el dispositivo en reposo ronda los 450 mA, aumentando considerablemente en los intervalos A y B.

El intervalo A se corresponde con la intensidad consumida en la recepción de cada paquete. Además, si se utilizan las técnicas de NC, se apreciará un mayor consumo, debido a que por cada paquete recibido es necesario comprobar si este es linealmente dependiente con alguno de los anteriores, y si lo es, descartarlo y esperar al siguiente.

El intervalo B se corresponde con la transmisión del ACK una vez recibidos todos los paquetes necesarios para obtener la información completa. Como se ha comentado anteriormente, el número de paquetes necesarios puede ser mayor que el de símbolos totales, debido a las dependencias lineales entre ellos.

Una vez comprendida la parte anterior, es importante conocer las diferentes configuraciones de NC seleccionadas durante la campaña de medidas. Por un lado, parte de los parámetros mencionados en el Capítulo 2 sobre las técnicas de NC se han mantenido estáticos, comenzando por el número de generaciones. Como el objetivo final del proyecto es averiguar el consumo energético debido a las técnicas de NC, incrementar el número de generaciones solo aumentaría dicho consumo linealmente ya que todas las generaciones contarían con los mismos parámetros de configuración. El siguiente parámetro estático es el tamaño de los símbolos y los paquetes. Cada símbolo está compuesto por 1416 bytes, y cada paquete por 1420 bytes. Como se ha explicado anteriormente, estos 4 bytes de diferencia se deben a la cabecera que se introduce en los paquetes con parámetros de configuración necesarios en los mismos.

Por otro lado, con el fin de obtener unos resultados relevantes, se han tomado medidas para diferentes valores de  $GF(2^q)$  y distinto número de paquetes por generación. Los valores seleccionados para variar el tamaño del campo finito han sido  $q = \{1, 4 \text{ y } 8\}$  mientras que el número de paquetes por generación se ha variado entre 1, 10, 20, 30, 40, 50, 100 y 200. De esta manera, con las tres configuraciones de  $q$  para cada uno de los tamaños de generación mencionados se han tomado medidas, tanto en transmisión como en recepción, siguiendo el procedimiento explicado en el Capítulo 3.

Para poder seguir dicho procedimiento, y conocer el consumo energético que introducen únicamente a las técnicas de NC, es necesario tomar muestras utilizando UDP de manera convencional y, una vez almacenadas, procesar todas las muestras con las diferentes configuraciones de NC mencionadas. Es necesario realizar todas las medidas dos veces para obtener el consumo energético tanto en transmisión como en recepción.

## 4.2 Consumo energético total en transmisión y recepción

Siguiendo todos los pasos explicados durante el trabajo y con las características expuestas en la sección anterior se han obtenido los siguientes resultados.

En la Tabla 4.1 se puede ver la intensidad media consumida por una Raspberry Pi transmitiendo con las distintas configuraciones mencionadas durante un segundo.

Tabla 4.1: Intensidad media consumida por una Raspberry Pi en una transmisión inalámbrica (mA)

Paquetes	NNC	q = 1	q = 4	q = 8
1	462.65	461.33	463.15	465.30
10	464.26	463.48	464.71	468.26
20	465.83	465.55	466.67	470.65
30	467.86	467.91	469.09	473.52
40	469.78	470.14	471.64	476.23
50	471.65	472.69	473.79	479.30
100	480.47	483.63	488.41	492.49
200	497.47	513.57	520.26	526.09

Así, como la fuente de alimentación proporciona 5 voltios continuos, la Tabla 4.2 recoge la energía consumida por el dispositivo durante la misma comunicación. Para ciertas configuraciones, los valores obtenidos con NC son menores que sin utilizarlo. Esto significa que el consumo debido a las técnicas de NC es prácticamente nulo y teniendo en cuenta la incertidumbre característica de un entorno real, se dan dichas situaciones. Se puede suponer que el consumo que introduce NC con tales configuraciones es prácticamente cero.

Además, en la Figura 4.2 aparecen los datos de la Tabla 4.1 representados gráficamente, pudiendo apreciar mejor la tendencia de consumo con las diferentes opciones de comunicación mencionadas.

Se puede observar que el consumo energético requerido es significativamente mayor cuando se utiliza el protocolo UDP combinado con técnicas de NC. Además, cuando aumenta el tamaño del cuerpo finito  $GF(2^q)$ , más intensidad requiere el dispositivo para transmitir la información.

Tabla 4.2: Consumo Energético de una Raspberry Pi en una transmisión inalámbrica (J)

Paquetes	NNC	q = 1	q = 4	q = 8
1	2.31325	2.30665	2.31576	2.32650
10	2.32129	2.31741	2.32353	2.34132
20	2.32917	2.32777	2.33333	2.35324
30	2.33930	2.33956	2.34549	2.36758
40	2.34890	2.35071	2.35821	2.38116
50	2.35823	2.36344	2.36891	2.39650
100	2.40233	2.41815	2.44204	2.46244
200	2.48737	2.56784	2.60132	2.63049

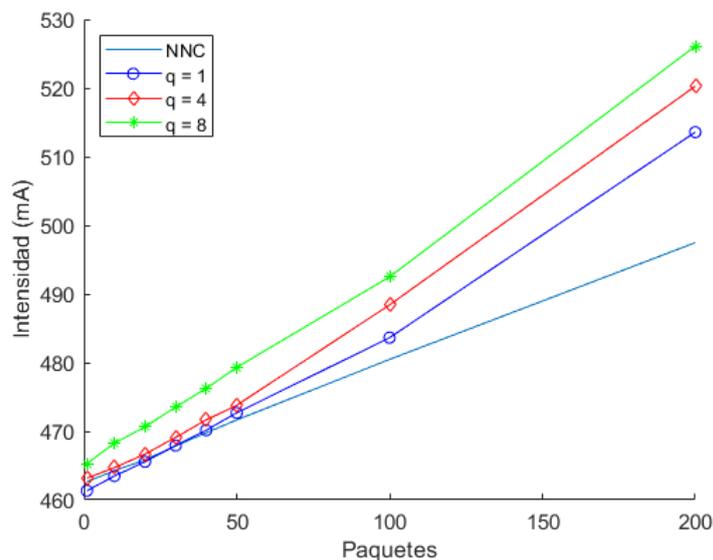


Figura 4.2: Intensidad media consumida por una Raspberry Pi en transmisión

Como el tamaño de los paquetes enviados es siempre el mismo, es lógico que cuantos más paquetes se envíen, mayor será el consumo, ya que se está enviando más información, es decir, más bytes de carga útil. Por ello, es interesante fijarse en los puntos donde el número de paquetes enviados coincide, cambiando únicamente el tamaño del cuerpo finito. En casi todos los casos se aprecia un aumento del consumo empleado en la transmisión, sobre todo a medida que el número de paquetes por generación es mayor, siendo este un escenario esperado, como se comentaba en el Capítulo 2.

Cuando se envían muy pocos paquetes por generación y el campo finito

es el mínimo, las operaciones de codificación de NC son muy simples, por lo que, teniendo en cuenta la incertidumbre propia de los entornos reales, no es extraño que el consumo en transmisión con estas características esté algo por debajo que el necesario cuando no se utilizan las técnicas de NC. Además, a medida que aumenta el número de paquetes en la generación, el crecimiento del consumo se mantiene lineal en la transmisión sin NC, mientras que cuando se utilizan estas técnicas, el aumento es ligeramente exponencial. Sin embargo, este crecimiento no es consecuencia únicamente de las técnicas de NC, sino también al aumento de la cantidad de bytes transmitidos.

Tabla 4.3: Intensidad media consumida por una Raspberry Pi en una recepción inalámbrica (mA)

Paquetes	NNC	q = 1	q = 4	q = 8
1	446.97	450.21	476.81	478.11
10	447.28	474.26	478.57	482.58
20	447.60	475.02	479.85	483.64
30	448.82	477.12	480.48	485.09
40	449.86	479.49	483.04	487.33
50	451.59	481.18	485.73	489.93
100	459.98	492.43	499.63	503.63
200	470.18	510.54	515.15	519.11

Tabla 4.4: Consumo Energético de una Raspberry Pi en una recepción inalámbrica (J)

Paquetes	NNC	q = 1	q = 4	q = 8
1	2.23486	2.25103	2.38406	2.39054
10	2.23642	2.37128	2.39282	2.41289
20	2.23801	2.37509	2.39927	2.41822
30	2.24411	2.38558	2.40242	2.42549
40	2.24931	2.39744	2.41522	2.43667
50	2.25796	2.40589	2.42866	2.44966
100	2.29990	2.46213	2.49819	2.51816
200	2.35090	2.55269	2.57577	2.59560

Por otro lado, las Tablas 4.3 y 4.4 recogen los datos de intensidad media y energía consumidos por una Raspberry Pi en recepción, respectivamente. Observando estos resultados y la Figura 4.3, que representa los valores de la Tabla 4.3, se aprecia un aumento del consumo energético a medida que aumenta el tamaño del campo finito.

Mientras que en transmisión se aprecia un crecimiento del consumo lineal en las comunicaciones sin NC y ligeramente exponencial cuando se utilizan estas técnicas a medida que aumenta el número de paquetes enviados, el dispositivo receptor incrementa su consumo energético de forma lineal, tanto en comunicaciones sin NC como empleando estas técnicas con las diferentes configuraciones utilizadas, siempre y cuando se alcancen unos paquetes por generación significativos, reduciendo así el efecto de la incertidumbre comentado previamente. Como se ha mencionado, este crecimiento también se debe al incremento de la cantidad total de bytes transmitidos. Sin embargo, a continuación se analiza la parte de este crecimiento debido únicamente a las técnicas de NC. La Figura 4.3 ya adelanta que el consumo debido a las técnicas de NC se ve acentuado en la recepción.

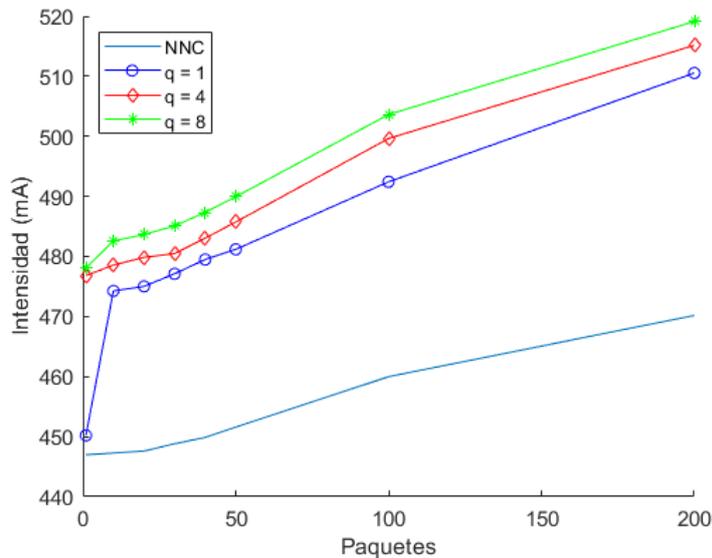


Figura 4.3: Intensidad media consumida por una Raspberry Pi en recepción

### 4.3 Consumo energético introducido por técnicas de Network Coding

A lo largo de este documento se ha mencionado en repetidas ocasiones que el objetivo final del trabajo es conocer el consumo energético adicional que introducen las técnicas de NC en una comunicación inalámbrica.

Tabla 4.5: Intensidad adicional consumida por una Raspberry Pi en una transmisión inalámbrica debido a las técnicas de NC (mA)

Paquetes	q = 1	q = 4	q = 8
1	-1.3200	0.5023	2.6507
10	-0.7749	0.4487	4.0051
20	-0.2818	0.8313	4.8121
30	0.0511	1.2365	5.6555
40	0.3637	1.8624	6.4523
50	1.0416	2.1361	7.6540
100	3.1636	7.9414	12.021
200	16.095	22.790	28.625

Tabla 4.6: Consumo energético adicional de una Raspberry Pi en una transmisión inalámbrica debido a las técnicas de NC (J)

Paquetes	q = 1	q = 4	q = 8
1	-0.0066	0.0025	0.0133
10	-0.0039	0.0022	0.0200
20	-0.0014	0.0042	0.0241
30	0.0003	0.0062	0.0283
40	0.0018	0.0093	0.0323
50	0.0052	0.0107	0.0383
100	0.0158	0.0397	0.0601
200	0.0805	0.1139	0.1431

Como ya se puede apreciar en las secciones anteriores de este capítulo, el incremento de dicho consumo es distinto en transmisión y recepción, siendo algo mayor en este último caso.

Para apreciar estas diferencias se han extraído los datos de energía consumidos únicamente por dichas técnicas. Así, en la Tabla 4.5 se muestran los valores de intensidad media consumidos debido a técnicas de NC en transmisión durante una comunicación inalámbrica, utilizando las mencionadas Raspberry Pi, mientras que la Tabla 4.6 recoge la energía consumida en Julios, calculada a partir de dicha intensidad.

Como se ha mencionado en la sección anterior, los valores negativos que aparecen en las configuraciones con tamaños de GF y número de paquetes por generación pequeños se pueden asociar a la incertidumbre propia de una plataforma real, por lo que se interpretan dichos valores como ceros.

En la Figura 4.4 aparecen representados los valores de la Tabla 4.5, donde se puede apreciar claramente un aumento del consumo requerido por la Raspberry Pi a medida que se aumenta el tamaño del campo finito.

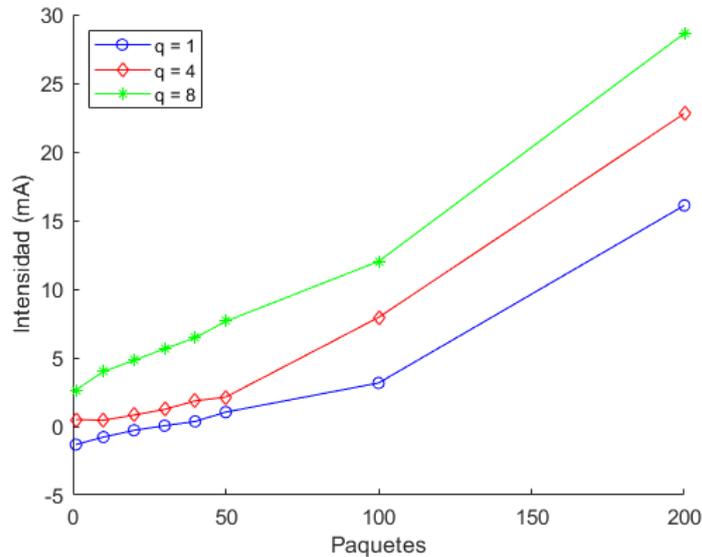


Figura 4.4: Intensidad media introducida por las técnicas de NC en transmisión

Sin embargo, el consumo en transmisión que introducen las técnicas de NC es despreciable para generaciones con un número de paquetes reducido, siempre y cuando el tamaño del campo finito no sea muy grande. El consumo en transmisión empieza a ser relevante cuando se aumenta el número de paquetes por generación, habiendo alcanzado en estas pruebas un valor máximo de 30 mA, obtenido con  $GF(2^8)$ , y una generación de 200 paquetes.

En la sección anterior se mencionó que el consumo energético seguía un crecimiento ligeramente exponencial con el aumento del tamaño de la generación. Sin embargo, dicho crecimiento podría estar relacionado con el incremento de la cantidad de bytes enviados. En la Figura 4.4 el crecimiento representado se debe únicamente a las técnicas de NC, por lo que se puede concluir que el consumo energético aumenta exponencialmente con el tamaño de la generación en transmisión.

Extrayendo el consumo introducido únicamente por las técnicas de NC en recepción, se obtiene la Tabla 4.7 que recoge todos los valores de intensidad medios obtenidos para las diferentes configuraciones utilizadas. En este caso no se aprecia ningún valor negativo, debido a que los consumos introducidos por NC son mayores en recepción. Mientras que en la transmisión se podían despreciar los valores para ciertas configuraciones, en recepción, la intensidad media consumida configurado el cuerpo binario y usando un paquete por

generación (configuración que introduce el menor consumo) supera los 3 mA.

Además, hay que tener en cuenta que esta configuración apenas introduce consumo adicional, debido a que no existe una matriz de codificación más simple. Si se apunta al siguiente valor de intensidad, que roza los 27 mA, se confirma que el consumo en recepción introducido por las técnicas de NC es considerablemente mayor, siendo comparable al mayor en transmisión observado durante este trabajo. Así, a partir de estos datos se ha obtenido el consumo energético de una Raspberry Pi en recepción debido a las técnicas de NC durante un segundo: Tabla 4.8.

Tabla 4.7: Intensidad adicional consumida por una Raspberry Pi en una recepción inalámbrica debido a las técnicas de NC (mA)

Paquetes	q = 1	q = 4	q = 8
1	3.2333	29.839	31.135
10	26.973	31.282	35.295
20	27.416	32.251	36.041
30	28.294	31.661	36.275
40	29.626	33.181	37.472
50	29.585	34.140	38.339
100	32.446	39.659	43.653
200	40.359	44.974	48.940

Tabla 4.8: Consumo energético adicional de una Raspberry Pi en una recepción inalámbrica debido a las técnicas de NC (J)

Paquetes	q = 1	q = 4	q = 8
1	0.0162	0.1492	0.1558
10	0.1349	0.1564	0.1765
20	0.1371	0.1613	0.1802
30	0.1415	0.1583	0.1814
40	0.1481	0.1659	0.1874
50	0.1479	0.1707	0.1920
100	0.1622	0.1983	0.2183
200	0.2018	0.2249	0.2447

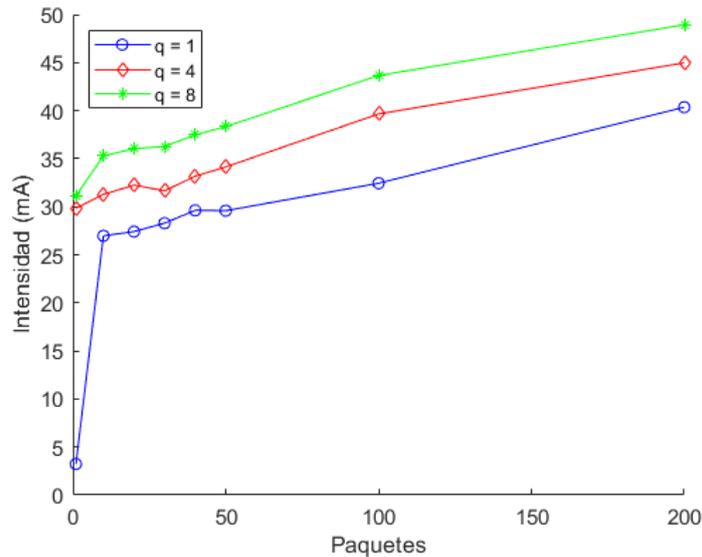


Figura 4.5: Intensidad media introducida por las técnicas de NC en recepción

A diferencia que en transmisión, donde el crecimiento del consumo sigue una tendencia exponencial, la energía adicional que requieren las técnicas de NC en recepción sigue un crecimiento lineal a medida que se aumenta el número de paquetes por generación, como se puede apreciar en la Figura 4.5. Por otro lado, como se adelanta en el Capítulo 2, el consumo energético de la Raspberry Pi aumenta a medida que lo hace el tamaño del campo finito empleado. Esto se debe al aumento de la complejidad de la matriz de codificación, requiriendo más recursos del sistema para su procesado.

Siguiendo con la diferencia de consumo que introducen las técnicas de NC en transmisión y recepción, la Figura 4.6 recoge tres gráficos en los que se compara directamente el consumo para los distintos tamaños de GF utilizados durante este trabajo. Para las tres configuraciones que se muestran en la figura, el consumo en recepción es considerablemente mayor que en transmisión, independientemente del número de paquetes por generación transmitido. Mientras que el consumo introducido por las técnicas de NC en transmisión es insignificante en configuraciones poco complejas, en recepción ronda los 30 mA.

Por otro lado, en esta misma figura se aprecia un acercamiento entre ambos consumos a medida que el tamaño de la generación aumenta. Esto concuerda con los crecimientos exponenciales y lineales en transmisión y recepción respectivamente explicados en secciones anteriores.

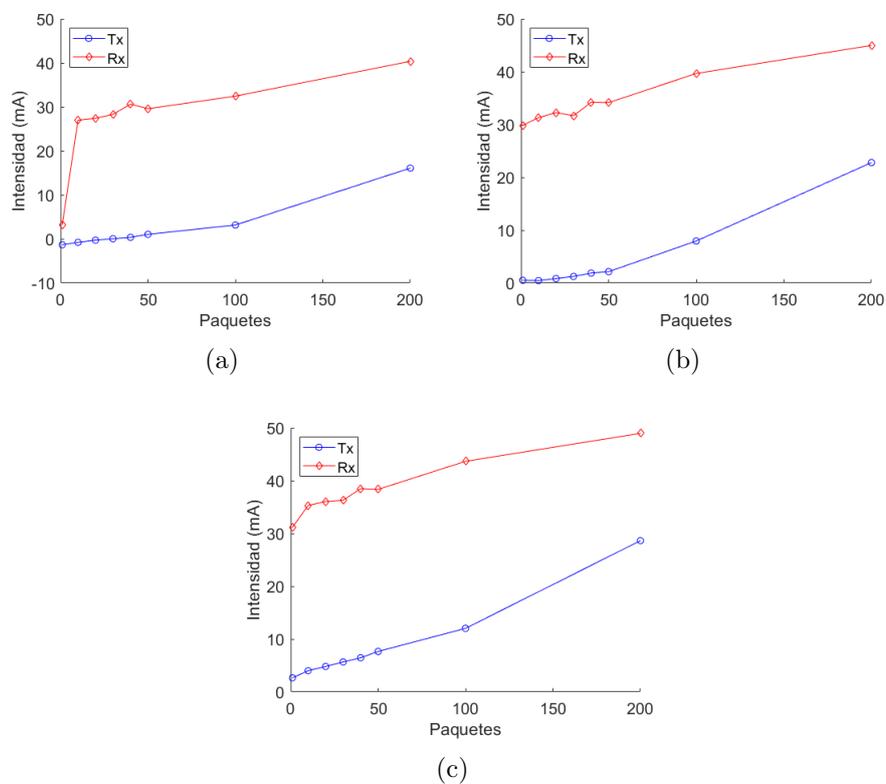


Figura 4.6: Comparativa de la intensidad requerida en transmisión y recepción para diferentes tamaños de campo finito. (a)  $q = 1$ , (b)  $q = 4$ , (c)  $q = 8$ .

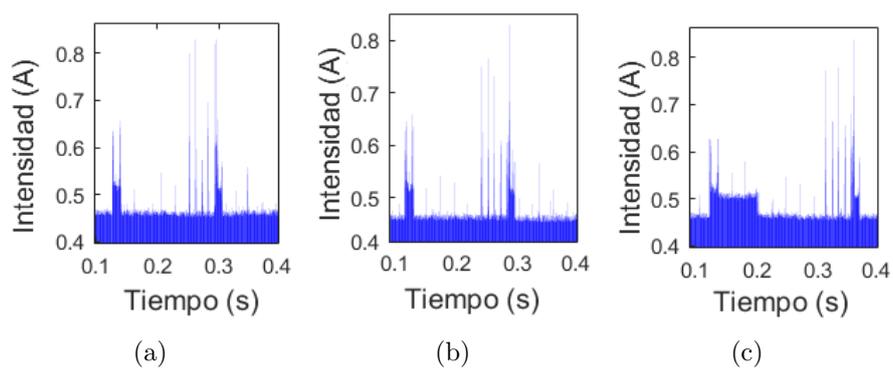


Figura 4.7: Intensidad (A) requerida en cada instante de tiempo (s) para diferentes tamaños de campo finito en transmisión. (a)  $q = 1$ , (b)  $q = 4$ , (c)  $q = 8$ .

Por último, la Figura 4.7 presenta la intensidad instantánea consumida en transmisión durante un segundo para las distintas configuraciones utilizadas en el trabajo.

Como se explica al principio de este capítulo, en la Figura 4.7 aparecen diferentes intervalos donde la intensidad consumida aumenta. Es interesante ver cómo, a medida que se incrementa el tamaño del campo finito, el intervalo A identificado en la Figura 4.1a también es mayor, destacando en la configuración con  $GF(2^8)$ . Esto se debe a que, como la complejidad de la matriz de codificación es mayor, es más costoso generar las combinaciones lineales de los paquetes nativos al construir los paquetes a enviar.

# Capítulo 5

## Conclusiones

### 5.1 Conclusiones

A lo largo de este documento se han presentado los resultados finales del consumo energético necesario en una comunicación inalámbrica utilizando las técnicas de NC, así como todo el proceso previo hasta llegar a los mismos.

La primera conclusión es que, efectivamente, las técnicas de NC introducen un consumo adicional en las comunicaciones inalámbricas. Sin embargo, dicho consumo está más acentuado en recepción, pudiendo incluso despreciarlo en transmisión cuando se utilizan las configuraciones más sencillas.

Como se muestra en el Capítulo 4, tanto en transmisión como recepción, los dispositivos involucrados en la comunicación inalámbrica precisan más energía cuando el tamaño del campo finito,  $GF(2^q)$ , aumenta. Esto se debe al incremento en la complejidad de la matriz de codificación, que exige más recursos para su procesado.

El otro factor principal valorado en este TFG es el tamaño de la generación, es decir, el número de paquetes enviados antes de recibir el ACK que confirma su correcta recepción. También se ha comprobado que un aumento de este tamaño incrementa el consumo energético tanto en transmisión como en recepción.

Por último, aunque todos los resultados obtenidos durante este trabajo

ponen de manifiesto un mayor consumo debido a NC en recepción que en transmisión, se ha observado que el crecimiento en ambas situaciones es distinto. Mientras que en recepción se mantiene un crecimiento lineal del consumo con el aumento del tamaño de la generación, en transmisión sube de forma exponencial. Por ello, parece que si se aumenta el número de paquetes en la generación por encima de los valores utilizados en este trabajo, el consumo adicional en transmisión podría alcanzar los resultados en recepción, incluso superarlos.

## 5.2 Líneas futuras

Con todos los resultados obtenidos en este TFG y la creación de la plataforma de medidas utilizada en el mismo, se abre un abanico de posibles trabajos futuros.

En primer lugar, puede ser interesante la automatización de la plataforma de medidas, ya que esta primera versión, hay muchas operaciones “manuales”. De esta manera se podrían recoger muchos más resultados en un intervalo de tiempo reducido, permitiendo abarcar objetivos más ambiciosos.

Por otro lado, en cuanto al consumo energético introducido por las técnicas de NC, se debería comprobar si los consumos en transmisión y recepción llegan a igualarse con el aumento del tamaño de la generación, corroborando la tendencia observada durante este trabajo.

Finalmente, se pueden variar otros parámetros de la configuración en las técnicas de NC además del tamaño del cuerpo finito y la generación, y observar el consumo introducido por ellos. También sería interesante explotar la plataforma para caracterizar algún otro tipo de soluciones NC, como Sparse Network Coding.

# Anexo A

## Función visualizar

```
function [] = Visualizar()

    clc, clear
    format long;

    %Lectura de las medidas almacenadas en la hoja de cálculo.
    str1 = 'C:\Users\David\Desktop\Medidas\NNC_10\DATA_1.csv';
    [num,txt] = xlsread([str1]);

    %Almacenar las medidas en una matriz y los parámetros necesarios en
    variables.
    MatrizS = num;
    Matriz_Size = size(MatrizS);
    Sample_Total = Matriz_Size(1,1)
    Sample_Interval = str2double(cell2mat(txt(2,2)))

    %Representar los valores en una gráfica.
    x = 0:Sample_Interval:((Sample_Total*Sample_Interval)-Sample_Interval);
    y = MatrizS(:,2)*(10^-8);

    figure(2)
    bar(x,y,'blue')

    xlabel('Tiempo (s)')
    ylabel('Intensidad (A)')
    title('Consumo Energético')

end
```

# Anexo B

## Función media

```
function [] = Media()

    clc, clear
    format long

    %Lectura de todas las hojas de cálculo con las medidas recogidas por el
    %multímetro.
    str1 = 'C:\Users\David\Desktop\Medidas\NNC_10';
    str2 = '\DATA_';
    str3 = '.csv';
    NumberA = '0';
    NumberB = '1';
    NumberC = '2';
    NumberD = '3';
    NumberE = '4';

    [num,txt] = xlsread ([str1 str2 NumberA str3]);
    [num1,txt1] = xlsread ([str1 str2 NumberB str3]);
    [num2,txt2] = xlsread ([str1 str2 NumberC str3]);
    [num3,txt3] = xlsread ([str1 str2 NumberD str3]);
    [num4,txt4] = xlsread ([str1 str2 NumberE str3]);

    %Almacenamiento de los datos en matrices de NumMuestras X 2.
    MatrizA = num;
    MatrizB = num1;
    MatrizC = num2;
    MatrizD = num3;
    MatrizE = num4;
```

```
%Cálculo del tamaño de cada matriz
MatrizA_Size = size(MatrizA);
Sample_TotalA = MatrizA_Size(1,1);
MatrizB_Size = size(MatrizB);
Sample_TotalB = MatrizB_Size(1,1);
MatrizC_Size = size(MatrizC);
Sample_TotalC = MatrizC_Size(1,1);
MatrizD_Size = size(MatrizD);
Sample_TotalD = MatrizD_Size(1,1);
MatrizE_Size = size(MatrizE);
Sample_TotalE = MatrizE_Size(1,1);

%Cálculo de la intensidad media consumida en cada caso.
A = (sum(MatrizA(:,2).*10^-8))/Sample_TotalA
B = (sum(MatrizB(:,2).*10^-8))/Sample_TotalB
C = (sum(MatrizC(:,2).*10^-8))/Sample_TotalC
D = (sum(MatrizD(:,2).*10^-8))/Sample_TotalD
E = (sum(MatrizE(:,2).*10^-8))/Sample_TotalE

Media = ((A+B+C+D+E) / 5)

end
```

# Bibliografía

- [1] P. G. Ortiz, *Opportunistic Network Coding over Wireless Networks*. PhD thesis, Universidad de Cantabria, 2018.
- [2] R. Ahlswede, Ning Cai, S. . R. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, July 2000.
- [3] S. . R. Li, R. W. Yeung, and Ning Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, Feb 2003.
- [4] R. Koetter and M. Medard, “An algebraic approach to network coding,” in *Proceedings. 2001 IEEE International Symposium on Information Theory (IEEE Cat. No.01CH37252)*, pp. 104–, June 2001.
- [5] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, “The benefits of coding over routing in a randomized setting,” in *IEEE International Symposium on Information Theory, 2003. Proceedings.*, pp. 442–, June 2003.
- [6] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Transactions on Information Theory*, vol. 52, pp. 4413–4430, Oct 2006.
- [7] A. M. González, “Experimental characterization of wi-fi and bluetooth interfaces in raspberry pi 3 devices,” 2017.