

# Simulador de Red Cognitiva

Enrique García Troyano

24 de mayo de 2016

## 1. Introducción

Este documento es una guía sobre el simulador realizado por la Universidad de Málaga bajo el proyecto *Eficiencia energética en redes inalámbricas cognitivas y de bajo consumo para un desarrollo sostenible* (NEREIDAS).

En el simulador desarrollado se definen una serie de parámetros de la red, como el número de usuarios primarios y cognitivos, las dimensiones de la red, las características de potencia de los nodos, y las características de propagación. En función de esos datos de entrada se simula el comportamiento de las transmisiones que lleva a cabo la red cognitiva frente a una determinada tasa de actividad en los nodos de la red primaria.

El simulador devuelve como salida la localización de los nodos, las conexiones existentes entre ellos, y para cada sesión de comunicaciones la ruta que ésta sigue, y la evolución temporal de los datos a través de la ruta. Esto se hace mostrando el estado de los enlaces de la ruta y de los buffers de los nodos durante el tiempo que dura la sesión. Además el simulador calcula algunos resultados de la transmisión como el throughput, la tasa de ocupación de los enlaces o el retardo para cada sesión.

La estructura del documento es la siguiente: en primer lugar se lleva a cabo una breve introducción sobre la radio cognitiva, posteriormente se introduce el simulador, explicando los algoritmos de encaminamiento implementados y la estructura funcional y de datos del mismo. Una vez explicado el funcionamiento del simulador, se presenta una simulación ejemplo, mostrando el resultado obtenido. Posteriormente se presentan una serie de resultados obtenidos sobre el comportamiento de la red frente a distintos parámetros de entrada.

Por último se presentan unas conclusiones y una serie de posibles ampliaciones a llevar a cabo en el simulador.

## 2. Radio Cognitiva

Debido a la gran importancia que han tomado las redes inalámbricas en la última década, en la que se han convertido en la principal tecnología utilizada en redes de acceso a Internet, las bandas de frecuencia utilizadas en este tipo de comunicaciones están alcanzando un nivel de saturación que obliga a buscar nuevas soluciones para la asignación de los recursos radio. En la actualidad, estos recursos se asignan siguiendo un esquema estático en el cual distintos grupos de usuarios tienen licencia para operar en determinadas bandas de frecuencias, de forma que cuando todas las bandas asignadas a un grupo

de usuarios están ocupadas, el siguiente usuario de ese grupo que quiera acceder al medio no podrá hacerlo, aunque existieran otros grupos de usuarios que no estuvieran utilizando todas las bandas que tienen asignadas. Esta situación genera, por un lado, usuarios que no pueden utilizar el medio, y por otro, bandas de frecuencias no utilizadas, lo que conforma un esquema de asignación de recursos ineficiente, que ha llevado a la situación actual en la que existe simultáneamente una saturación en el número de bandas disponibles, y un aprovechamiento de las bandas asignadas muy bajo.

La radio cognitiva se basa en un nuevo paradigma sobre la asignación de los recursos radio, según el cuál cuando un grupo de usuarios se queda sin frecuencias porque todas las que tenía asignadas se están utilizando, el siguiente usuario que quiera utilizar el medio sea capaz de percibir si existen bandas asignadas a otros usuarios que no están siendo utilizadas. En ese caso, el usuario que quiere acceder al medio, podría utilizar dicha banda, siempre y cuando algún usuario con licencia original no la necesite. En caso de que un usuario con licencia quiera utilizarla, el nodo que la estaba ocupando debe abandonarla inmediatamente. A los usuarios con licencia se les denomina usuarios *primarios* y a los que aprovechan las bandas no utilizadas por los primarios se les denomina usuarios *cognitivos* o *secundarios*.

Por lo tanto, un nodo cognitivo debe ser capaz de sensor su entorno radioeléctrico y adaptar sus transmisiones al mismo. De esta forma es posible aumentar la eficiencia espectral de las redes inalámbricas, ya que cada nodo puede conocer las frecuencias que se encuentran disponibles en un determinado instante y adaptar su funcionamiento a las condiciones existentes. En otras palabras, la radio cognitiva permite la implementación de estrategias de asignación de recursos radio de forma dinámica, sin empeorar la calidad de servicio que se presta a aquellos usuarios que tienen una cantidad de recursos previamente asignada, lo que redundaría en un aprovechamiento mucho mayor de las bandas de frecuencia y conforma una posible solución al problema de saturación de las frecuencias utilizadas en redes inalámbricas.

A pesar de que los sistemas de radio cognitiva se han investigado desde hace varios años, aún no se han implementado estos sistemas de manera comercial debido a que requiere superar múltiples retos, entre los que se pueden mencionar: el desarrollo de técnicas para monitorizar el estado del medio y la actividad de los nodos primarios de forma más eficiente, técnicas de modulación para minimizar las interferencias que los nodos cognitivos causan sobre los primarios, algoritmos que tomen decisiones en base a la información que les llega de los distintos nodos de la red, lo que puede llevar a redes más o menos distribuidas y a distintos niveles de cooperación entre nodos, las técnicas de control de acceso al medio y la asignación de frecuencias entre usuarios cognitivos o la elaboración de algoritmos de encaminamiento que tengan en cuenta las frecuencias disponibles en cada nodo a nivel local.

Este tipo de redes también suponen una oportunidad para aumentar la eficiencia energética de las comunicaciones inalámbricas, ya que al poseer los nodos de estas capacidades cognitivas para medir su entorno radioeléctrico, se hace posible optimizar también el consumo energético total de la red, a través de redes multi-salto donde las distancias de los enlaces sean mucho más pequeñas y por tanto las potencias de transmisión necesarias sean menores. Sin embargo, para poder implementar capacidades cognitivas en los nodos, se requiere un elevado nivel de procesamiento en cada nodo de la red, lo que puede considerarse un factor limitante debido al elevado consumo energético que esto puede suponer, sobre

todo si se tiene en cuenta que estos nodos pueden ser dispositivos móviles que operen con baterías. Esta limitación ha motivado un gran número de estudios sobre el uso de nodos *relays* (de relevo) que ayuden a transmitir los mensajes de la red, y que podrían suponer una infraestructura fija, que limitaría el consumo energético de los usuarios finales.

### 3. El Simulador

El modelo de propagación considerado es el de un canal con pérdidas de propagación por espacio libre con shadowing log-normal. Es decir que si la potencia transmitida es un valor  $P_t$  la potencia recibida será

$$P_r = P_t(d^{-\alpha})10^{-\frac{\sigma_x^2}{10}}$$

donde  $\alpha$  representa las pérdidas de propagación por espacio libre y  $\sigma_x^2$  el factor de shadowing que modela las irregularidades del entorno físico.

En el simulador se asume que los nodos cognitivos detectan de forma perfecta cuándo el primario está inactivo y cuándo se encuentra transmitiendo datos. El simulador consiste en un bucle en el que cada iteración representa a un instante de tiempo, es decir que el tiempo está dividido en *time slots*, a los que se llama *quantums*. En cada quantum (en cada iteración del bucle de simulación) los nodos cognitivos deben comprobar el estado de las bandas de los primarios y de los otros cognitivos para saber qué bandas pueden utilizar y qué bandas están prohibidas. En caso de que dos nodos cognitivos quieran utilizar una misma frecuencia en un mismo time slot, la contienda se resuelve de forma aleatoria, y el nodo que obtiene el medio sólo lo abandona cuando termina de transmitir o cuándo el primario lo expulsa. Una vez comprobadas las bandas disponibles, se realizan las transmisiones correspondientes a cada time slot, para después comprobar si algún nodo ha terminado de transmitir sus datos y si algún primario se ha activado.

De manera previa al bucle de simulación temporal, se generan las sesiones de datos a enviar entre los nodos cognitivos. Estas sesiones indican el nodo origen y el nodo destino y la cantidad de datos que se quiere transmitir de uno a otro. Una vez generadas las sesiones se calcula el encaminamiento de éstas entre los nodos cognitivos y con los resultados del algoritmo de encaminamiento se simulan las transmisiones en el bucle temporal.

#### 3.1. Algoritmos de Encaminamiento Utilizados:

En el simulador se han implementado dos algoritmos de encaminamiento diferentes. El primero de ellos cuantifica cada enlace de la red con un peso y luego elige el camino con peso mínimo entre Origen y Destino. El cálculo de los pesos de los enlaces se hace teniendo en cuenta la capacidad del enlace y la disponibilidad de los enlaces de estar disponible, que se calcula en función de su dependencia con los nodos primarios y de la actividad que presentan los diferentes primarios. Este algoritmo es similar al implementado en [3], pero mejorado, ya que el cálculo de la probabilidad del enlace no se hace de forma aleatoria como en el estudio citado, sino que se hace teniendo en cuenta un área de influencia para cada primario y un factor de actividad que indica la probabilidad de que el primario esté activo y que el nodo cognitivo es capaz de estimar mediante la observación del entorno radioeléctrico. De esta manera el nodo cognitivo sabe cuáles son los usuarios primarios

más activos. Además el número de frecuencias disponibles en la red cognitiva es igual al número de nodos existentes en la red primaria.

El segundo algoritmo utilizado es el presentado en [1]. Dicho algoritmo elige el camino con menor número de saltos, teniendo en cuenta la actividad de los nodos primarios. El algoritmo evita aquellos caminos que tengan una alta dependencia a primarios con un factor de actividad elevado. Además también tiene en cuenta un coste de cambio de frecuencia, por lo que considera antes aquellos caminos en los que no será necesario realizar transmisiones en distintas frecuencias sino que podrá utilizarse una misma banda en todo el camino.

### 3.1.1. Camino de Retardo Mínimo modificado.

Este algoritmo selecciona el camino con retardo mínimo entre el origen y destino, teniendo en cuenta la probabilidad de estar disponible de los enlaces que atraviesa la ruta, lo que depende de la actividad de los usuarios primarios. Para hacer esto le asigna un peso a cada enlace y aplica el algoritmo de Dijkstra, que calcula el camino de menor peso entre un nodo A y otro nodo B dentro de un grafo. El resultado obtenido dependerá del criterio según el cuál se asignan los pesos a los enlaces. En este sentido se tiene en cuenta, por un lado, la calidad del enlace a través del retardo y por otro, la probabilidad de estar disponible a través de la dependencia del enlace a los usuarios primarios. Este algoritmo se ha elaborado tomando elementos de [3] y de [1]. De [3] se toma la idea de asignarle un peso a cada enlace según el retardo y calcular el camino con menor retardo a través del algoritmo de Dijkstra. De [1] se ha tomado la idea de definir una zona de influencia y un factor de actividad para cada nodo primario, de manera que cada nodo cognitivo elabore un vector de influencias que indica cómo le afecta cada nodo primario. De esta forma se tiene en cuenta la actividad de los nodos primarios en la disponibilidad de los enlaces y se evitan aquellos caminos que estén muy congestionados.

El retardo de un enlace es el tiempo que éste tarda en transmitir un bit, es decir, el inverso de la capacidad. La Capacidad se calcula para un modelo de propagación con pérdidas por espacio libre y Shadowing. Por lo tanto, la capacidad de un enlace será

$$C = BW \cdot \log_2(1 + \gamma)$$

donde BW representa el ancho de banda del canal y  $\gamma$  representa la Relación Señal a Ruido (SNR) en el receptor, es decir

$$\gamma = \frac{Pot_{Tx} \cdot d^{(-\alpha)} \cdot 10^{(N_{shad}/10)}}{BW \cdot N_o}$$

La probabilidad de que una banda de frecuencias de un enlace esté ocupada se calcula como el máximo entre las influencias de los nodos que conforman el enlace a dicha frecuencia. La probabilidad de que el enlace esté ocupado se calcula como una media de las probabilidades de que las distintas frecuencias estén ocupadas. Por lo tanto la probabilidad de que el enlace esté disponible será:

$$P_{i \rightarrow j} = 1 - \left( \frac{1}{N_p} \sum_{f_p=1}^{f_p=N_p} \max\{I_i^{f_p}, I_j^{f_p}\} \right)$$

Donde  $I_i^{fp}$  y  $I_j^{fp}$  se refieren a la influencia del primario p sobre los nodos cognitivos i y j respectivamente. Esta influencia es nula si los nodos están fuera de la zona de influencia del primario y es igual al factor de actividad del primario en caso contrario.

Si ninguno de los nodos del enlace está afectado por el primario, entonces el enlace tiene probabilidad de estar disponible uno (Esta probabilidad se refiere a los primarios, podría ocurrir que el enlace estuviera prohibido porque lo está utilizando otro usuario cognitivo).

El algoritmo de retardo mínimo modificado tiene en cuenta la calidad de los enlaces, pero no toma su decisión en función de este factor únicamente sino que también tiene en cuenta la disponibilidad que tendrá el enlace debido a la actividad de los nodos cognitivos. Esto hace que sea muy importante que el nodo cognitivo tenga una buena estimación de la actividad del primario. Para implementar una buena estimación de la actividad del primario podrían elaborarse métodos más complejos que no dependan únicamente de la actividad observada en los últimos N time slots, sino que tenga en cuenta otros factores como los tiempos de duración medios de las transmisiones y de los silencios, de forma que el cognitivo pueda adaptar su estimación lo más rápido posible.

Al implementar el algoritmo no se ha tenido en cuenta la comunicación entre los nodos cognitivos sobre el estado de la red, y se supone que cada nodo tiene conocimiento total sobre el estado de la misma. En la práctica, esta solución requiere de un ente centralizado que toma las decisiones de encaminamiento e informa a los nodos mediante un canal de control común, el nodo centralizado conoce el estado de la red debido a las mediciones que los nodos han hecho y que le han transmitido a través del canal común. Este canal común será un canal de baja frecuencia (no requiere de tasas de transmisión elevadas y presenta un alcance mayor). Cada nodo puede actualizar su información de forma periódica o mediante un sistema de alarmas, de forma que cuando note algún cambio significativo lo notifique al ente que gestiona la red.

### 3.1.2. Algoritmo GYMKHANA

Este algoritmo se ha implementado de forma idéntica a como se define en [1]. Al contrario que el anterior, es un algoritmo distribuido, esto presenta la ventaja de que para implementarlo no es necesario un canal de control común, ni un nodo centralizado que gestione la asignación de frecuencias, sino que la información sobre el encaminamiento y el estado de la red se va propagando de unos nodos a otros, mediante el envío de paquetes *Route Request* y *Route Reply*, lo que aumenta la necesidad de procesamiento de los nodos y aumenta la cantidad de tráfico generado en la red, sobre todo para redes con un alto nivel de conectividad en las que el cálculo del camino óptimo tiene un mayor coste debido a la mayor cantidad de caminos posibles.

En este algoritmo de encaminamiento se tienen en cuenta dos factores: Por un lado el número de saltos que debe atravesar el camino, y por otro, la dependencia de los nodos del camino a la actividad de los primarios, que influye de dos formas diferentes: en el número de cambios de frecuencia que deben producirse en los distintos enlaces y en la probabilidad de que los enlaces estén disponibles. El algoritmo asigna un peso de coste de cambio de frecuencia. Teniendo en cuenta estos factores, se calcula una función de utilidad para cada camino y se elige aquél cuya función de utilidad es máxima. Esta función de utilidad se calcula mediante la elaboración de un grafo Virtual y la evaluación de la matriz Laplaciana  $\mathbb{L}$  de dicho grafo virtual.

El algoritmo de encaminamiento consta de dos fases: en la primera se hallan todos los caminos posibles, mediante la retransmisión desde el origen de una *Route Request* (RREQ). Cada RREQ incluye información sobre los nodos por los que va pasando y los vectores de influencia de cada uno de esos nodos. Cuando un nodos recibe la RREQ comprueba que su identificador no esté en la lista, si su identificador aparece significa que se ha formado un bucle y la descarta. Si no la había recibido, agrega su información a la RREQ y la retransmite a todos sus vecinos. De esta manera el nodo destino recibirá una RREQ por cada camino posible entre Origen y Destino, y en cada una de ellas tendrá información sobre la influencia de los primarios sobre los nodos cognitivos de la ruta.

Cuando el nodo destino ha recibido todas las RREQ, empieza la segunda fase del algoritmo: el nodo Destino debe seleccionar la Ruta Óptima. Para hacerlo, crea un grafo virtual por cada RREQ recibida. Dicho grafo virtual consiste en un grafo tipo rejilla donde cada fila representa los distintos nodos que forman la ruta a una misma frecuencia, y cada columna las distintas frecuencias en un mismo nodo de la ruta. Por lo tanto un enlace vertical en el grafo virtual representa un cambio de frecuencia y un enlace horizontal representa la transmisión a una frecuencia determinada de un nodo de la ruta al siguiente. El peso de los enlaces horizontales depende de la influencia de los primarios sobre los nodos que conforman el enlace a la frecuencia correspondiente, mientras que los enlaces verticales tienen un peso  $\alpha$  constante que representa el coste de cambiar de frecuencia. El grafo virtual será un grafo de  $(N_p \times (H_k + 1))$  elementos, donde  $N_p$  es el número de frecuencias y  $H_k$  es el número de enlaces de la ruta.

Una vez formado el grafo virtual, se evalúa su Matriz Laplaciana  $\mathbb{L}$ , y se genera la función de Utilidad. La matriz Laplaciana de una Matriz  $(N \times M)$  es una matriz  $(N \cdot M \times N \cdot M)$ . La laplaciana se define como la Matriz de Grados  $\mathbb{D}$  menos la Matriz de Adyacencias  $\mathbb{A}$  ( $\mathbb{L} = \mathbb{D} - \mathbb{A}$ ), donde  $\mathbb{D}$  es una matriz diagonal en la que cada elemento de la diagonal indica la suma de los pesos de los enlaces adyacentes al nodo correspondiente, y  $\mathbb{A}$  es una Matriz binaria en la que un elemento  $(i,j)$  de la matriz es igual a cero si los nodos  $i$  y  $j$  no están conectados y es igual al peso del enlace en caso contrario.

La matriz Laplaciana de un grafo bidireccional es simétrica y la suma de todas sus filas y de todas sus columnas es igual a cero. Además la Matriz Laplaciana permite obtener información sobre la conectividad del grafo. En concreto, se cumple que el primer autovalor de  $\mathbb{L}$  en un grafo bidireccional es siempre nulo ( $\lambda_1 = 0$ ) y el número de autovalores nulos es igual al número de componentes conexas del grafo. Por lo tanto si  $\lambda_2 = 0$ , el grafo  $G$  no está conectado. Al segundo autovalor de  $\mathbb{L}$ ,  $\lambda_2$  se le llama *conectividad algebraica*. Se ha demostrado que la conectividad algebraica supone una medida de la estabilidad y robustez en modelos de redes complejos [2] [1].

Para calcular la función de utilidad de cada camino, en primer lugar se genera el grafo virtual de la ruta suponiendo que no hay usuarios primarios, se calcula la Laplaciana de dicho grafo virtual y su conectividad algebraica, que para este caso se denota como  $\lambda_2^{clear}$ . Posteriormente se calcula la Laplaciana de la ruta con los usuarios primarios y se calcula su conectividad algebraica  $\lambda_2$ . Por último se tiene en cuenta el número de enlaces  $H_k$  que conforman la ruta para calcular la función de Utilidad del camino  $k$ , que se define como:

$$U_k = \frac{(\lambda_2)_k}{(\lambda_2)_k^{clear}} \cdot \frac{1}{H_k}$$

El nodo destino escoge aquella ruta cuyo valor de  $U_k$  es máximo. Esta métrica tiene en cuenta el número de enlaces de la ruta, la influencia de los primarios y la cantidad de

cambios de frecuencia que se deben llevar a cabo a lo largo del camino. A partir de la definición de  $U_k$  resulta inmediato que si sólo existe un canal de comunicación y la ruta no está afectada por los primarios, el algoritmo elige el camino con el mínimo número de saltos (camino más corto).

Al contrario que en el caso anterior, para este algoritmo si se ha implementado el intercambio de información entre los nodos sobre el camino elegido. En el simulador se implementa la primera fase que consiste en hallar todos los caminos posibles a través de la retransmisión de RREQ's, posteriormente se implementa la evaluación de cada uno de los caminos hallados en el nodo destino y la elección de aquél con función de utilidad máxima. Por último se simula también la confirmación del camino desde el nodo Destino hacia los otros nodos de la ruta mediante la retransmisión de una *Route Reply* (RREP) en sentido inverso a los nodos de la ruta elegida. Estos nodos guardan en memoria la información sobre a qué nodo deben retransmitir los datos de la sesión.

Un problema que presenta este algoritmo en la práctica es que cuando aumenta el número de nodos cognitivos, los posibles caminos aumentan de forma exponencial, lo que lleva a una cantidad muy elevada de RREQs y además a una gran necesidad de cálculo en el nodo destino. En la práctica, al realizar las simulaciones con el algoritmo implementado en *Matlab*, a partir de un número de nodos cognitivos superior a 20 el tiempo de simulación se hace extremadamente elevado.

Debido a este motivo, como solución general se adopta el primer algoritmo propuesto.

## 3.2. Estructura del Simulador

### 3.2.1. Estructura de Datos

En el simulador se definen cinco clase diferentes. Estas son:

1. **Clase *Nodo***: Es una clase padre. Existen dos clases que heredan de *Nodo*, estas son: *NodoPrimario* y *NodoCog*. Los atributos de *Nodo* son los siguientes:
  - *PosX*: Coordenada X de la posición del nodo.
  - *Posy*: Coordenada Y de la posición del nodo.
  - *Tipo*: Indica si se trata de un nodo primario (Tipo=1) o de un nodo cognitivo (Tipo=2).
  - *ID*: Identificador del *Nodo*. Dos nodos de distinto tipo (primario y cognitivo) pueden tener el mismo identificador, pero dos nodos del mismo tipo nunca tendrán un identificador igual.
  - *PotTx*: Indica la potencia que transmite el nodo.
  - *Sensibilidad*: Indica la sensibilidad del nodo.
2. **Clase *NodoPrimario***: Hereda de la clase *Nodo*, por lo tanto tiene todos los atributos de dicha clase aparte de sus atributos propios. Cada objeto de esta clase representa a un nodo primario de la red. Los atributos propios de *NodoPrimario* son:
  - *VecinosCognitivos*: Nodos Cognitivos que están dentro de la zona de influencia del primario.

- *FactorActividad*: Probabilidad de que el primario esté transmitiendo. La simulación de la actividad de los primarios tiene como objetivo obtener una *TasaOcupacion* que tienda a este valor.
- *Estado*: Indica si el primario transmite (Estado=1) o no (Estado=0) en un time slot.
- *Transicion*: Indica si el estado del primario ha cambiado con respecto al time slot anterior.
- *QuantumsActivo*: Número de time slots que el primario ha estado transmitiendo. Permite calcular la tasa de ocupación del primario.
- *QuantumsInactivo*: Número de time slots que el primario ha estado sin transmitir. Permite calcular la tasa de ocupación del primario.
- *TasaOcupacion*: Tasa de ocupación del primario. Indica la actividad que ha presentado el primario hasta el instante actual. Es igual al número de Quantums Activo entre el número de Quantums inactivo.
- *ProbEvaluacion*: Se utiliza para generar la actividad de los primarios. Es la probabilidad de evaluar la tasa de ocupación del nodo primario y comprobar si debe cambiarse el estado del nodo para que esta se acerque al Factor de Actividad. Esta probabilidad se incrementa en cada iteración con un valor *increm*. Cuando se realiza una evaluación se reinicia esta probabilidad a 0 y se repite el proceso.
- *MargenOcupacion*: Margen de error permitido entre *TasaOcupacion* y *FactorActividad*. Si estos dos valores presentan un error mayor que *MargenOcupacion*, entonces se cambia el estado del nodo para que se reduzca el error.
- *HistorialActividad*: Vector donde se guarda la actividad que va a seguir el nodo primario durante cada instante del bucle de la simulación de transmisiones.

3. **Clase *NodoCog***: Esta clase también hereda de la clase *nodo*, por lo que también tiene los atributos de dicha clase, además de los suyos propios. Representa un nodo cognitivo de la red. Sus atributos propios son:

- *EstadoTx*: Vector de longitud  $N_p$ . Indica las transmisiones del nodo en las distintas bandas de frecuencia. Un valor igual a -1 indica que la transmisión está inhabilitada. Un valor igual a 0 indica que la transmisión está disponible y un valor igual a 1 indica que está transmitiendo.
- *EstadoRx*: Equivalente al anterior, pero referido a la recepción en lugar de la transmisión. Si un nodo transmite a una frecuencia, su recepción a esa frecuencia está inhabilitada y viceversa.
- *Vecinos*: Lista de los nodos cognitivos que están conectados con el nodo actual.
- *DatosPendientes*: Indica para cada sesión los bits que le quedan por transmitir al nodo. (Vector de longitud `Num_Sesiones`).
- *NodoDatosPendientes*: Indica para cada sesión cuál es el siguiente nodo al que debe enviar los datos el nodo actual. (Vector de longitud `Num_Sesiones`).



- *SesionEnTx*: Indica la sesión que se está transmitiendo en cada banda. (Vector de longitud  $N_p$ ).
- *VecinosPrimarios*: Lista de nodos primarios frente a los que el nodo cognitivo es susceptible de un desalojo, debido a que está en su zona de influencia.
- *Conectividad*: Numero de nodos cognitivos conectados al nodo, es decir número de enlaces adyacentes al nodo.
- *Buffer*: Indica la cantidad de bits que tiene el nodo en buffer para cada sesión en un momento dado. Un nodo no puede retransmitir los bits que recibe si no ha recibido el Transport Block correspondiente por completo. Es decir si no ha recibido los *TamBloque* bits que forman ese bloque de datos.
- *HistBuffer*: Guarda el estado de los buffers en cada instante de la simulación. Se utiliza para almacenar y presentar los resultados de la simulación.
- *VectorInfluencias*: Vector donde se guarda la influencia que tiene cada primario sobre el cognitivo. Es decir, el factor de actividad del primario si el cognitivo está en su zona de influencia, o cero si no es así.
- *BufferRREQ*: Buffer de mensajes tipo RREQ. Se utiliza para el encaminamiento de las sesiones mediante el algoritmo Gymkhana.

4. **Clase Sesión:** Esta clase representa las sesiones de comunicación entre nodos cognitivos. Es decir que representa las comunicaciones que se simulan. En cada objeto sesión se guarda información relacionada con el uso de las frecuencias, los tiempos de transmisión de cada nodo, los nodos activos en un instante dado, la ruta que sigue la sesión. Los atributos de esta clase son:

- *ID*: Identificador de la Sesión.
- *Estado*: Indica el estado de la sesión (0: No ha empezado a transmitir. 1: está transmitiendo. 2: Ya ha finalizado).
- *Origen*: Identificador del Nodo Origen de la sesión.
- *Destino*: Identificador del Nodo Destino de la sesión.
- *CantidadDatos*: Cantidad de Bits que quiere transmitir del nodo Origen hacia el nodo Destino.
- *Ruta*: Vector con los identificadores de los nodos de la ruta que debe atravesarse para llegar de Origen a Destino.
- *BandasEnlaces*: Vector de longitud igual al número de enlaces activos en la sesión. Cada elemento indica la frecuencia que está utilizando el enlace correspondiente en ese instante (-1 si está desalojado. Si el enlace ha finalizado su transmisión se queda con el valor de la ultima frecuencia utilizada).
- *t\_IniTxNodos*: Indica el instante en el que empieza a transmitir cada uno de los nodos de la ruta.
- *t\_FinTxNodos*: Indica el instante en el que deja de transmitir cada uno de los nodos de la ruta.
- *TiempoInicio*: Indica el instante en el que inicia la sesión.

- *TiempoFin*: Indica el instante en el que la sesión finaliza las transmisiones.
- *NodosActivos*: Vector que indica en cada iteración los nodos de la ruta que están activos en ese instante (Transmitiendo o Recibiendo Datos).
- *Finalizada*: Indica si la sesión ha finalizado (1) o no (0).
- *TamBloque*: Cantidad de bits que conforman un Bloque de Transporte a nivel de enlace. Un nodo intermedio no puede reenviar bits que correspondan a un bloque de transporte que no se ha recibido por completo.
- *Desalojados*: Indica los desalojos de frecuencia existentes en un instante dado en la sesión. Cada desalojo simboliza un enlace que estaba transmitiendo datos de la red cognitiva y tuvo que abandonar la banda porque el primario accedió a la misma. Un desalojo es una matriz de una columna y cuatro filas que indica el nodo origen, el nodo destino, la banda de frecuencia y el instante en el que ocurrió el desalojo. El atributo Desalojados Es una Matriz con un número de columnas igual al número de desalojos existentes en ese instante. Si no hay ningún desalojo será una matriz vacía.
- *NumDesalojos*: Número de desalojos de bandas de frecuencia que se han producido a lo largo de la transmisión de los bits de la sesión en cualquiera de los enlaces de la ruta.
- *HistorialBuffers*: Guarda la información del estado de los buffers en cada iteración. Esto permite presentar los resultados una vez concluida la simulación.

5. **Clase *ResultadosSesion***: Esta clase se utiliza para guardar información sobre la simulación en las distintas iteraciones y poder calcular el comportamiento final de la red. Sus atributos son:

- *TiempoEnlacesTransmitiendo* Representa la cantidad de tiempo que están transmitiendo los enlaces de la sesión.
- *TiempoEnlacesDesalojados* Representa la cantidad de tiempo que están desalojados los enlaces de la sesión, debido a que no tienen frecuencias disponibles para transmitir.
- *TiempoEsperaAsignacionBanda* Tiempo que está en enlace esperando para empezar a transmitir datos.
- *HistorialBandasEnlaces* Guarda el historial del estado de los enlaces.
- *TiempoSesion* Tiempo de duración de la sesión.
- *Throughput\_ Util* Indica los Bits de la sesión divididos por el tiempo que dura la sesión. Es el throughput útil en transmitir una cantidad de información del origen al destino.
- *Throughput\_ Total* Indica la cantidad de bits cursada por la red entre el tiempo que dura la sesión. Es igual al número de enlaces multiplicado por el throughput útil, si hay más enlaces la red debe cursar más cantidad de datos.
- *TasaOcupacionEnlaces* Indica la tasa de ocupación de los enlaces, o dicho de otra manera la eficiencia con que se utilizan los recursos radio.

### 3.2.2. Estructura Funcional

El simulador está compuesto por tres bloques funcionales básicos. El primero de ellos está asociado con la topología de la red a simular. En él se definen las dimensiones de la red, se sitúan los nodos, se definen los enlaces existentes entre nodos cognitivos en función de la distancia radioeléctrica a la que estos se encuentran y se asigna un peso a cada enlace relacionado con la capacidad del mismo y la probabilidad de estar disponible. En el segundo bloque se definen parámetros propios de la comunicación que se va a simular. En primer lugar se define la actividad de los nodos primarios generando un vector de actividad para cada nodo primario. Este vector indica cuándo transmite o no el nodo primario, y será un dato de entrada para el siguiente bloque. El motivo de generar la actividad de los nodos primarios en este bloque es que, de cara a la red cognitiva la actividad de los nodos primarios es un parámetro de entrada, por lo tanto para poder realizar diferentes simulaciones de la red cognitiva frente a unas mismas condiciones, es necesaria una misma actividad de los nodos primarios. Por otra parte, se definen las sesiones de comunicación de los usuarios cognitivos que se van a simular. Cada sesión se define mediante: Nodo Origen, Nodo Destino y Cantidad de Datos a transmitir. Finalmente, una vez definida la sesión en este bloque se calcula una ruta para la comunicación entre el nodo Origen y el Nodo Destino. En el tercer y último bloque se realiza la simulación de la red cognitiva. Este bloque consiste en un bucle temporal, en el que cada iteración simboliza un slot temporal. En cada iteración se comprueba el estado de los nodos primarios definido en el bloque anterior para comprobar los enlaces disponibles. En función de los enlaces disponibles y de las necesidades de comunicación definidas entre los nodos cognitivos se realizan las transmisiones correspondientes en cada bucle de simulación temporal hasta que las comunicaciones entre usuarios cognitivos hayan finalizado.

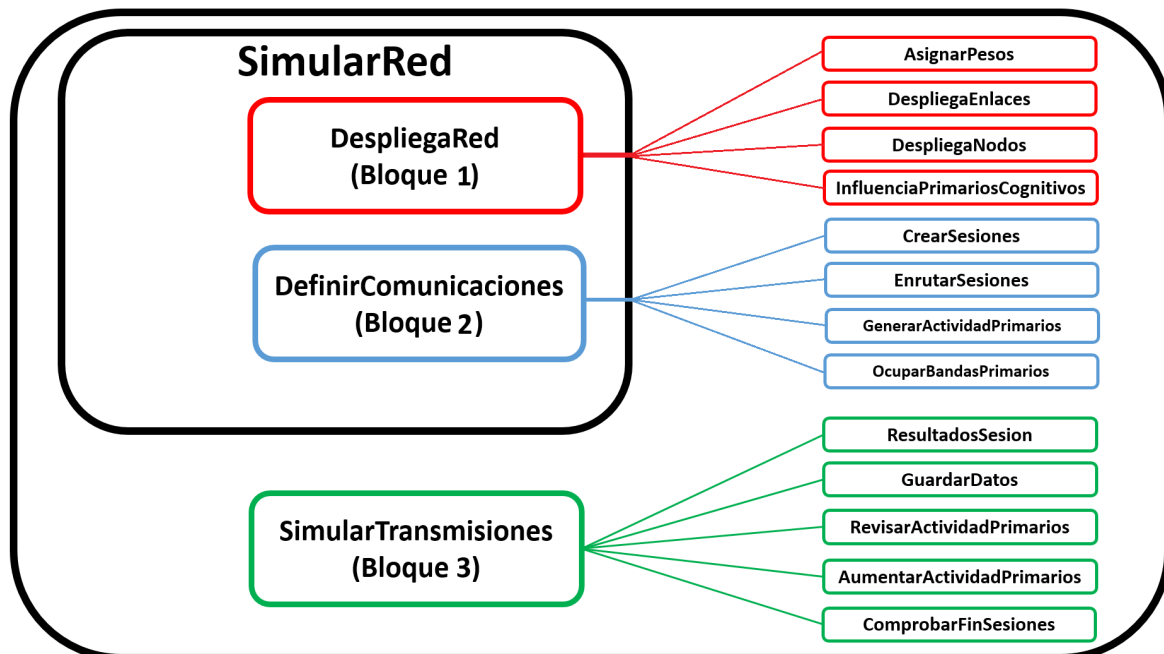


Figura 1: Esquema de Funciones del Simulador.

## Bloque 1:

**Función Principal:** *DespliegaRed.m* (Llamada desde el script *SimularRed*).

### **Parámetros de entrada:**

- XMAX: Dimensión en el eje X de la Red.
- YMAX: Dimensión en el eje Y de la Red.
- Np: Número de Nodos primarios.
- Ns: Número de Nodos cognitivos.
- FactoresAct: Valor de los Factores de Actividad de los nodos primarios. El Factor de actividad es la probabilidad de que el primario transmita. (Vector de longitud Np).
- PotTx: Potencia transmitida por los nodos cognitivos.
- Sens: Sensibilidad de los nodos cognitivos.
- PotPrim: Potencia transmitida por los nodos primarios.
- SensPrim: Sensibilidad de los nodos primarios.
- Alfa: Coeficiente de pérdidas de propagación.
- Fact\_shad: Coeficiente de Shadowing.
- BW: Ancho de Banda de los canales.
- No: Densidad Espectral de ruido.

### **Descripción:**

En función de las dimensiones de la red y del número de nodos se crea una rejilla. Se sitúa un único nodo en una posición aleatoria dentro de cada casilla de la rejilla. Si el número de nodos no se pudo distribuir de forma exacta en la rejilla, los nodos sobrantes se sitúan de forma aleatoria sobre toda el área de la red. Una vez situados los nodos se calcula la distancia radioeléctrica entre cada uno de los nodos cognitivos y los que estén a una distancia radioeléctrica menor a un umbral, es decir aquellos en los que la potencia recibida sea mayor que la sensibilidad, se conectan.

Posteriormente se calcula para cada nodo primario cuáles nodos cognitivos están dentro de su área de influencia, es decir, aquellos nodos que al transmitir generaría una interferencia mayor a la tolerable en el primario. Cada nodo cognitivo debe saber si está en el área de influencia de algún primario. Si un nodo cognitivo no está dentro del área de influencia de un primario, la influencia de dicho primario sobre el cognitivo será nula, si se encuentra dentro del área de influencia, entonces la influencia del primario sobre el cognitivo será igual al factor de actividad del primario. Por último para cada enlace entre nodos primarios se asigna un peso. Este peso será el inverso de la capacidad del enlace

ponderada por la probabilidad de que el enlace esté disponible. Es decir que un peso pequeño indica que el enlace es bueno y pesos elevados indican enlaces de mala calidad.

## **Bloque 2:**

**Función Principal:** *DefinirComunicaciones.m* (Llamada desde el script *SimularRed*).

### **Parámetros de entrada:**

- Num\_Sesiones: Número de sesiones de comunicación entre nodos cognitivos que se van a simular.
- LongDatos: Cantidad de bits de cada sesión de comunicación entre nodos cognitivos.
- T\_fin: Instante hasta el que se va a definir una comunicación para los nodos primarios. Si la transmisión de las comunicaciones entre nodos cognitivos superase este tiempo habría que aumentar el vector de actividad de los nodos primarios.
- Quantum: Valor del time slot, será el tiempo que representa cada iteración en el bucle de simulación del bloque 3. El vector de actividad de los nodos primarios se genera desde  $t=0$  hasta  $t=t\_fin$  con este paso.
- VectorNodosPrim: Vector donde se guarda la información de cada nodo primario.
- VectorNodosCog: Vector donde se guarda la información de cada nodo cognitivo.
- MatrizConexiones: Matriz con la información sobre los enlaces existentes en la red.

**Descripción:** Genera la simulación de la actividad de los nodos primarios, que será utilizada en el siguiente bloque a la hora de simular las transmisiones entre nodos cognitivos. Crea las Sesiones de comunicación entre los nodos cognitivos, y calcula el encaminamiento para cada una de estas sesiones creadas.

## **Bloque 3:**

**Script Principal:** *SimularTransmisiones.m*

### **Parámetros de entrada:**

- Quantum: Valor del Time Slot. Tiempo que representa una iteración del bucle.
- TamBloque: Valor del *Transport Block*: Mínima cantidad de bits que se pueden transmitir a nivel de enlace. Un nodo no puede retransmitir bits al siguiente nodo si no ha recibido una cantidad de bits mayor o igual a TamBloque.
- VectorNodosPrim: Vector de nodos primarios. Contiene la información generada en los bloques anteriores sobre los nodos primarios.
- VectorNodosCog: Vector de nodos cognitivos. Contiene la información generada en los bloques anteriores sobre los nodos cognitivos.

- **VectorSesiones:** Vector de Sesiones. Contiene la información generada en los bloques anteriores sobre las sesiones.
- **MatrizConexiones:** Matriz de Conexiones o enlaces. Contiene la información generada en los bloques anteriores sobre las conexiones entre nodos cognitivos.

**Descripción:** Consiste en un bucle en el que cada iteración supone un instante de tiempo de la simulación. Dentro de cada iteración el simulador realiza las siguientes funciones:

1. Comprobar si en el instante actual debe activarse alguna sesión. Cada sesión tiene un tiempo de activación y si el instante es mayor o igual a ese tiempo la sesión debe activarse.
2. Comprobar si para una sesión activa debe iniciarse una nueva transmisión. Es decir que un nodo que sólo estaba recibiendo datos deba empezar a transmitir al siguiente. Esto se hace cuando el nodo tiene un número de datos mayor al *Transport Block* del nivel de enlace.
3. Guarda los datos sobre el estado actual de la simulación.
4. Transmite los datos que correspondan en cada iteración. Estos datos dependen tanto de los datos que debería transmitir un nodo, como de si puede transmitirlos, es decir, de los enlaces disponibles que está limitada por la actividad de los primarios y de otros nodos cognitivos.
5. Comprueba si un nodo ha terminado sus transmisiones relacionadas con una sesión determinada.
6. Comprueba si una sesión ha finalizado.
7. Comprueba que no se haya alcanzado el tiempo hasta el que se definió comunicación en los nodos primarios. Si se alcanza este tiempo se aumenta el tiempo de actividad de los nodos primarios para poder seguir simulando.
8. En cada bucle una vez que hay transmisiones activas se comprueba si ha habido cambio en el estado de los nodos primarios y si por lo tanto se debe desalojar a algún nodo cognitivo que estuviera transmitiendo. También se comprueba lo contrario: si algún nodo cognitivo que estuviera desalojado por un primario puede reasignarse a la misma o a otra banda de frecuencias.

## 4. Ejemplo de una simulación

A continuación se muestran los resultados obtenidos para una simulación con los siguientes parámetros de entrada:

Número de usuarios primarios:  $N_p = 4$ . Número de usuarios secundarios:  $N_s = 10$ .

Dimensiones de la red:  $E_{jex} = 20$ .  $E_{jey} = 20$ .

Potencia transmitida:  $PotTx = 1$ ,  $PotPrim = 1$ .

Sensibilidad de los Nodos:  $Sens = 10^{-2}$ ,  $SensPrim = 10^{-3}$ .

Ancho de banda:  $BW = 10^3$ . Nivel de Ruido:  $No = 10^{-5}$ .  
Coeficiente de pérdidas de propagación:  $alfa = 2,5$ . Nivel de Shadowing:  $fact\_shad = 7dB$ .  
Número de sesiones cognitivas:  $Num\_sesiones = 2$ .  
Datos a transmitir en cada sesión:  $LongDatos = 10^4$ .  
Tipo de encaminamiento:  $TipoRout = 1$  (Algoritmo de retardo mínimo modificado).  
Factores de Actividad de los primarios:  $FA = 0,5$ .

La dependencia de los cognitivos con los primarios obtenidos en la simulación es la siguiente:

Primario 1: Afecta a los nodos cognitivos: 1, 2, 3, 4, 5, 6, 7, 9, 10.  
Primario 2: Afecta a los nodos cognitivos: 3, 6, 9.  
Primario 3: Afecta a los nodos cognitivos: 1, 4, 5, 7, 8, 9, 10.  
Primario 4: Afecta a los nodos cognitivos: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

En la figura 2 se representa la localización de los nodos dentro de la red y las conexiones existentes entre los nodos cognitivos. Los nodos cognitivos se representan con un triángulo azul y cada nodo primario se representa con un asterisco de distinto color. Se han desplegado 10 nodos cognitivos y 5 nodos primarios, por lo tanto existen 5 posibles bandas de frecuencia para la red cognitiva, estas bandas estarán disponibles según la actividad de los nodos primarios que se muestra en la figura 5.

La figura 3 muestra la simulación de la sesión número 1, que tiene como origen el nodo 7 y como destino el nodo 6. En ella se muestra la cantidad de bits en los buffers de los nodos de la sesión y el estado de los enlaces durante la transmisión, el estado de los enlaces se representa mediante barras cuya altura y color indican la banda de frecuencia que se está utilizando. Una misma frecuencia se representa con el mismo color en esta figura que los asteriscos en la figura 2 y que la actividad del primario correspondiente mostrada en la figura 5. Por otra parte, cuando el enlace ha sido desalojado, se representa con un valor igual a -1 y con barras de color negro.

La ruta elegida en la sesión 1 es  $7 \rightarrow 4 \rightarrow 6$ . Es decir que existen 3 nodos y 2 enlaces diferentes en la simulación de la sesión. Con la dependencia de los cognitivos con los primarios indicada anteriormente se comprueba que el primer enlace no depende del primario 2, por lo que estos dos nodos pueden utilizar la frecuencia 2 sin preocuparse de la actividad del primario debido a que la interferencia que causan sobre él está por debajo del valor umbral. Por este motivo, el primer enlace se transmite a la frecuencia 2 de forma ininterrumpida. El segundo enlace de la sesión está influenciado por todos los primarios, ya que no existe ningún primario para el cual tanto el nodo 4 como el nodo 6 no sean dependientes. Debido a este motivo, el segundo enlace de la sesión va cambiando de frecuencias, utilizando aquellas que están disponibles en un instante dado y abandonándolas cuando el nodo primario correspondiente necesita la banda.

La sesión 2 tiene su origen en el nodo 6 y su destino en el nodo 8. En la figura 4 se observa su simulación. En esta sesión la ruta elegida para transmitir los datos es  $6 \rightarrow 3 \rightarrow 5 \rightarrow 8$ . El primer enlace de la ruta no depende de la actividad del nodo primario 3, por lo tanto transmite siempre a esta frecuencia de forma ininterrumpida. Además el

último enlace de la sesión no depende del nodo primario 2, por lo que también transmite de forma ininterrumpida a esta frecuencia. El primer enlace de la sesión 1 y el último enlace de la sesión 2 pueden transmitir simultáneamente a la misma frecuencia porque los transmisores de cada enlace no están conectados con los receptores del otro y por lo tanto no se generan interferencias significativas. Por último, el enlace intermedio de la sesión 2 sí depende de todos los primarios y por eso va ajustando su banda de transmisión a los enlaces que tiene disponible en cada instante en función de la actividad de los nodos primarios y de los otros nodos cognitivos.

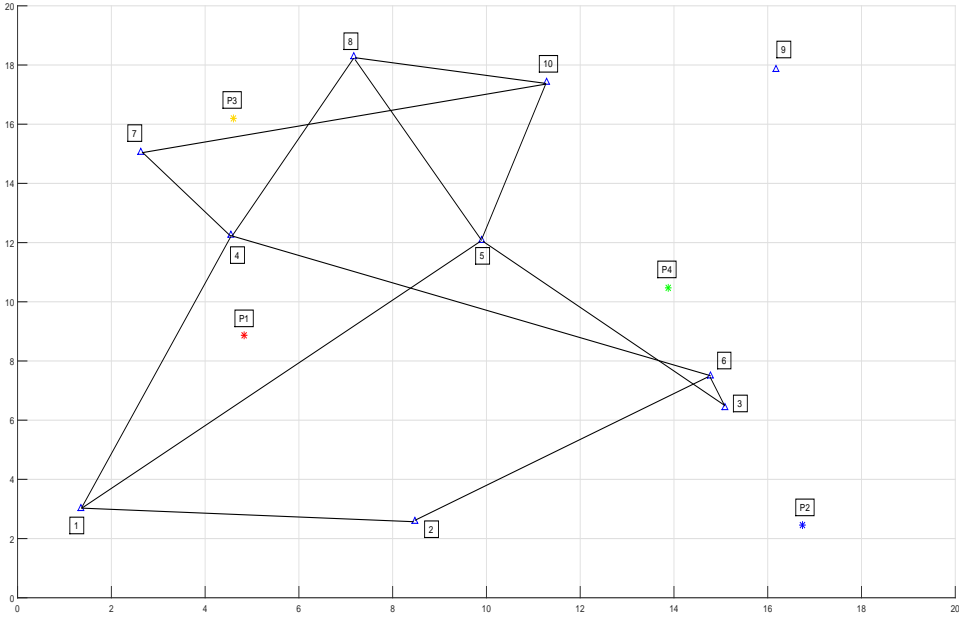


Figura 2: Localización de los nodos en una simulación.



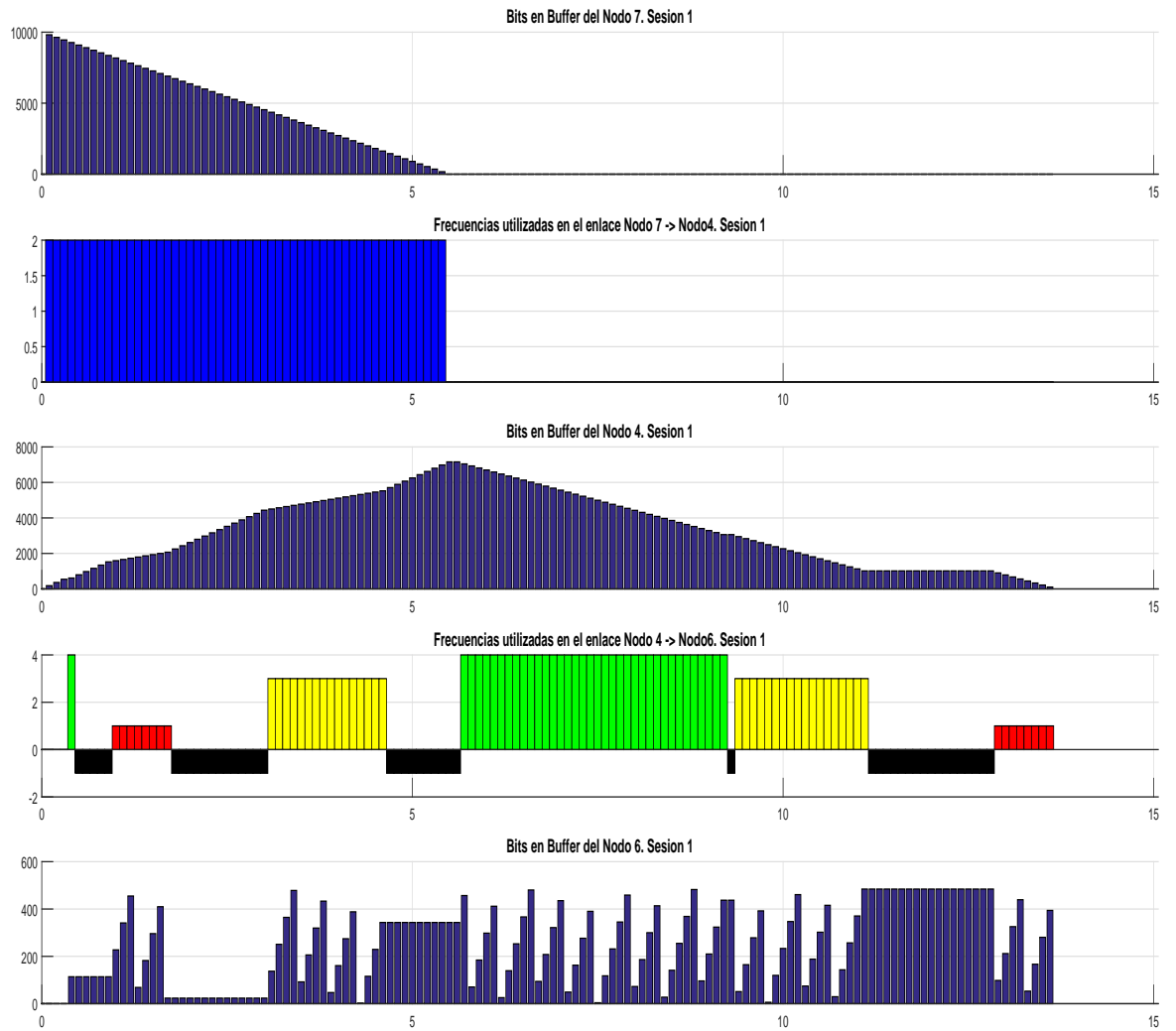


Figura 3: Estado de los buffers de los nodos y estado de los enlaces durante la transmisión de la sesión 1, entre el nodo 7 y el nodo 6.

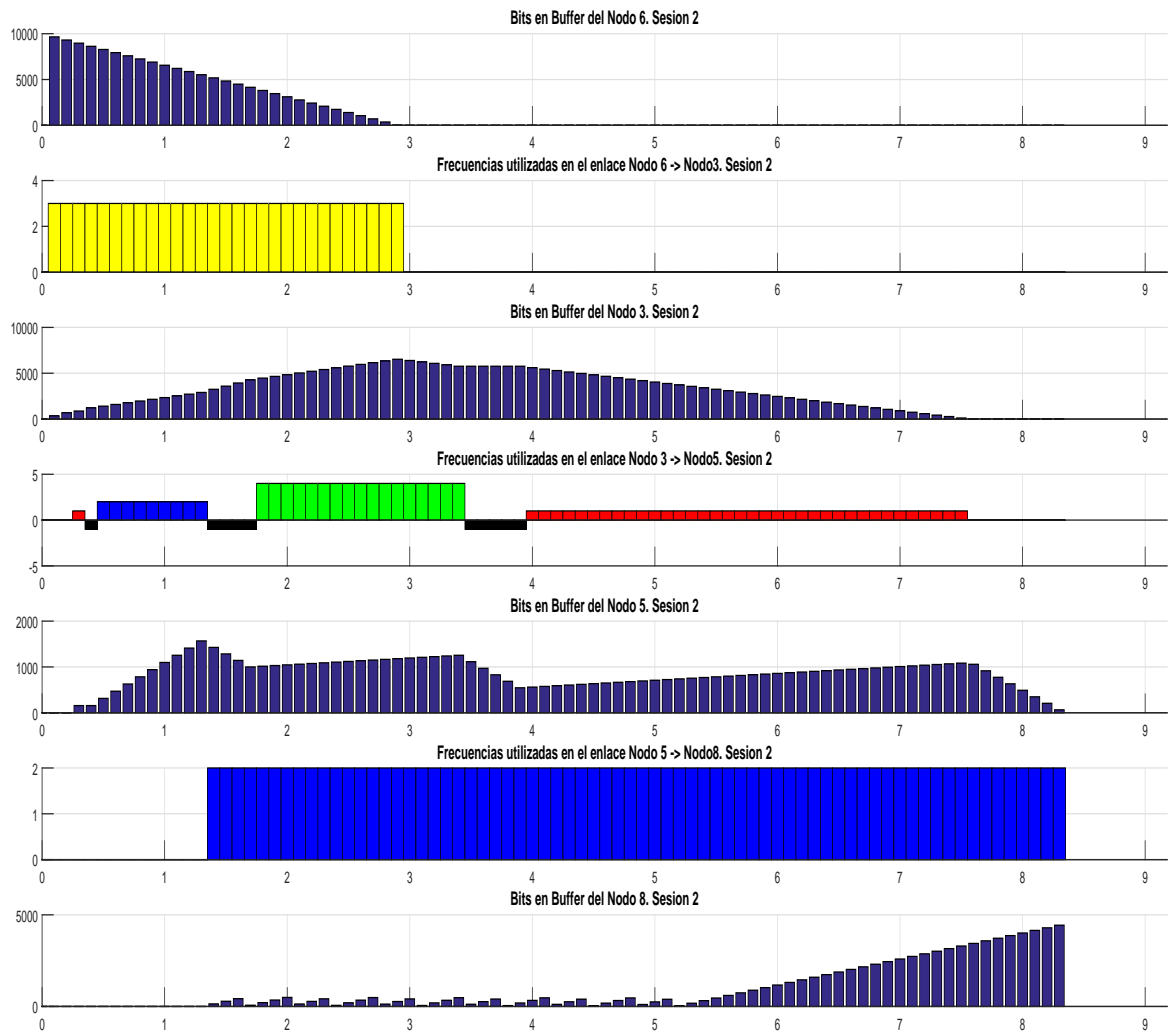


Figura 4: Estado de los buffers de los nodos y estado de los enlaces durante la transmisión de la sesión 1, entre el nodo 6 y el nodo 8.

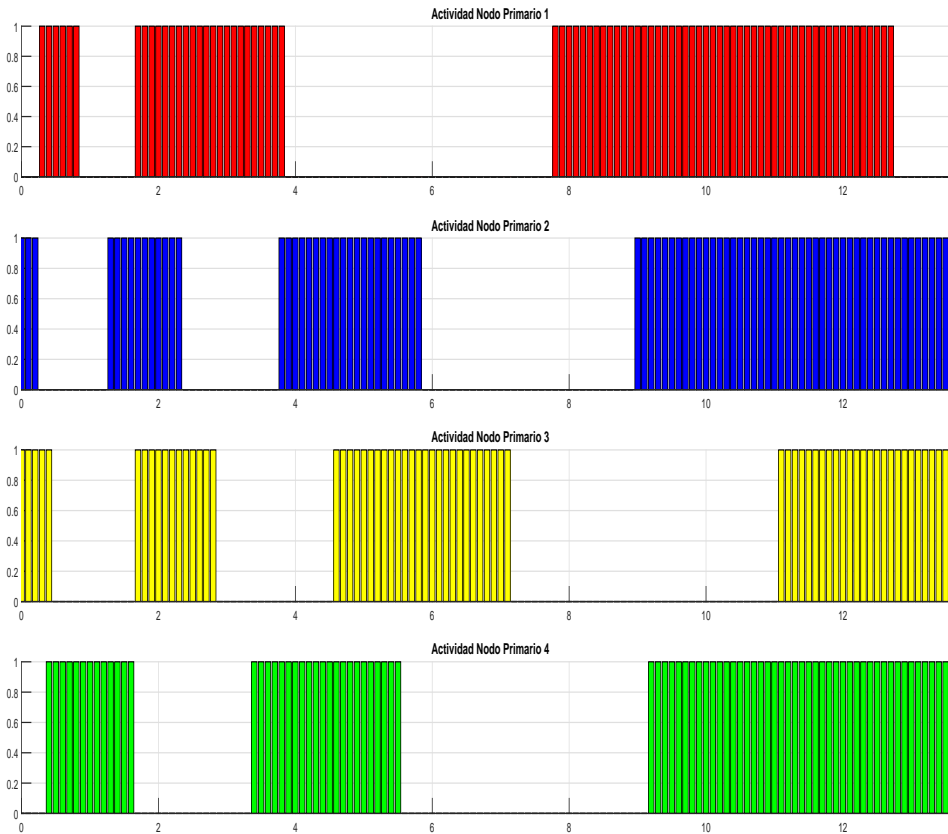


Figura 5: Actividad de los nodos primarios durante la simulación.

## 5. Comportamiento de la red en función de algunos parámetros de entrada

A continuación se estudia la influencia de algunos parámetros de entrada sobre el comportamiento de la red. Para ello se realiza un número de simulaciones para cada valor del parámetro a estudiar y se promedian los resultados obtenidos. Cada simulación implica generar una nueva posición de los nodos, generar de nuevo los enlaces existentes y para esos nodos y enlaces correspondientes simular las transmisiones.

De esta manera se puede observar como afecta la variación de un parámetro al comportamiento de la red.

### 5.1. Influencia de $N_s$ :

Para estudiar la influencia del número de nodos cognitivos se realizan 100 simulaciones para cada valor de  $N_s$ . Es importante destacar que cuando la simulación no genera ningún enlace, se repite hasta que se haya generado al menos un enlace entre dos nodos. Debido

a ese motivo, cuando se define un valor de  $N_s$  muy pequeño para unas dimensiones de la red determinadas, hay que repetir varias veces la simulación hasta que se produzca una simulación en la que la posición de los nodos esté lo suficientemente cerca como para que se pueda crear un enlace.

Para cada simulación los parámetros de entrada son:

Dimensiones de la red:  $XMAX = 20, YMAX = 20$

Número de primarios:  $N_p = 5$

Factores de actividad de los primarios:  $FA = 0,5$

Número de sesiones:  $Num\_Sesiones = 4$

Longitud de datos de cada sesión:  $LongDatos = 5000$

Potencia transmitida:  $Pot = 1, PotPrim = 1$

Sensibilidad de los nodos:  $Sens = 0,01, SensPrim = 10^{-3}$

coeficiente de pérdidas de propagación:  $alfa = 2,5$

Quantum de tiempo:  $quantum = 0,1$ .

En primer lugar cabe destacar que cuando se realizan las simulaciones de las sesiones, los nodos origen y destino se toman de forma aleatoria y se comprueba que el nodo origen y el nodo destino dispongan de una ruta existente, por lo tanto, cuando  $N_s$  es pequeño, las sesiones de comunicación serán todas entre los mismos nodos o tendrán alguna diferencia, pero por lo general utilizarán los mismos enlaces, mientras que cuando  $N_s$  es elevado, la probabilidad de que dos sesiones necesiten utilizar el mismo enlace en el mismo instante se reduce, ya que las sesiones serán entre nodos diferentes con una gran probabilidad.

En los resultados se observa como para valores elevados de  $N_s$ , al aumentar el número de nodos cognitivos aumenta el throughput útil o lo que es equivalente, se reduce el retardo. Esto ocurre porque al aumentar el número de nodos cognitivos manteniendo el número de sesiones de comunicación, aumenta la cantidad de recursos disponibles de la red cognitiva, ya que al desplegarse más nodos también se crean más enlaces. Además la mejora que un aumento de  $N_s$  produce es cada vez menor, debido a que para valores muy elevados ya existen enlaces entre los nodos cognitivos de sobra, y lo que está limitando el comportamiento de la red es la actividad de los nodos primarios. Es por eso que la tasa de ocupación no mejora más allá de un valor por mucho que aumente  $N_s$ .

El comportamiento que se produce cuando  $N_s$  es pequeño se debe a que para valores muy bajos de  $N_s$  ( $N_s=3$ ) sólo existen conexiones en aquellos casos en que los dos nodos cognitivos estén muy cerca el uno del otro y en este caso sólo existirá una o dos conexiones en la red, por lo que los enlaces serán de una longitud muy pequeña y de muy buena calidad, además la longitud de las rutas será mucho más pequeña. Cuando este número aumenta un poco, las sesiones ya son más largas y por lo tanto ya deben concantarse transmisiones en varios enlaces, es por ello que el throughput útil baja y el retardo sube para valores de  $N_s=10$  ó  $12$  respecto a  $N_s=3$  ó  $5$ . En la práctica lo que ha ocurrido es que la red se ha hecho más grande porque antes se tenía sólo una pequeña zona de la red conectada. Por último para valores pequeños de  $N_s$  se obtiene una tasa de ocupación más baja en los enlaces, esto se debe a que como existen muy pocos enlaces, todas las sesiones tienen que utilizar los enlaces existentes y por lo tanto las transmisiones de la red cognitiva deben esperar unas por otras. Es por eso que el retardo cuando  $N_s$  es muy grande es mejor que para estos valores, pues ya no está limitado por los otros cognitivos, sino que está limitado únicamente por los primarios.

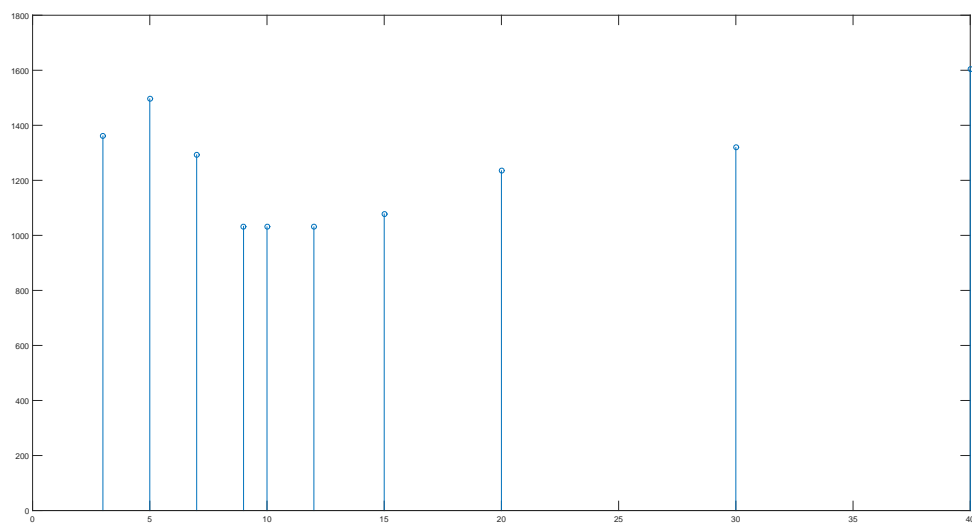


Figura 6: Througput Util (bps/duración de la sesión) en función del Número de nodos Cognitivos.

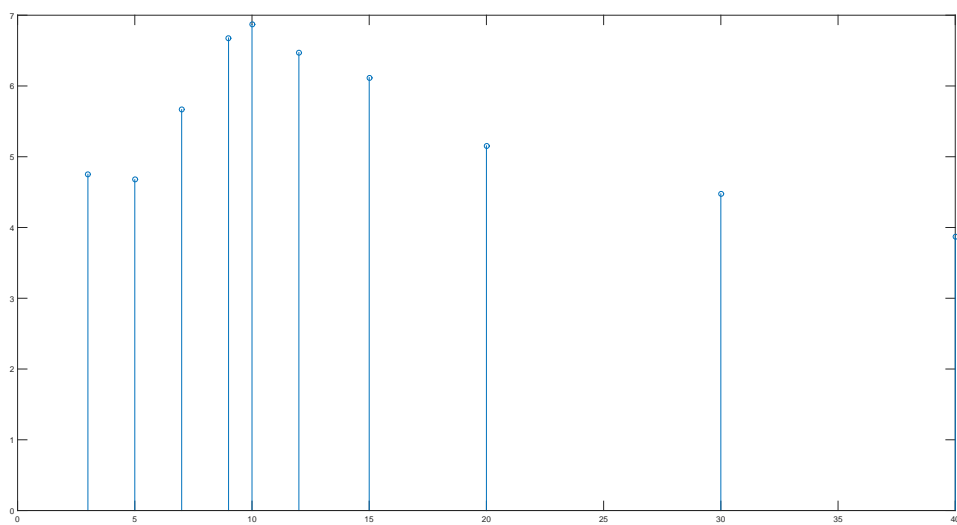


Figura 7: Retardo en función del Número de nodos Cognitivos.

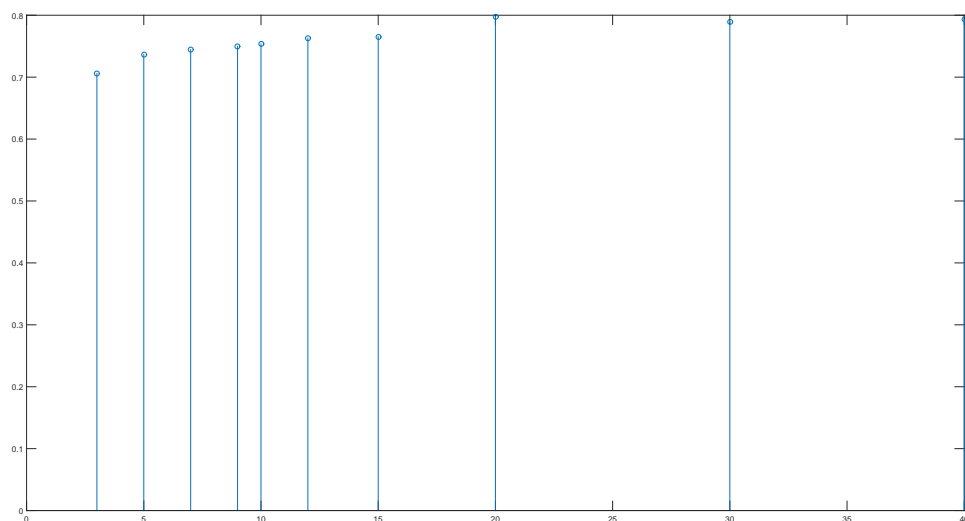


Figura 8: Tasa de ocupación de los enlaces en función del Número de nodos Cognitivos.

## 5.2. Influencia de $N_p$ :

Para estudiar la influencia del número de nodos primarios se realizan 50 simulaciones para cada valor de  $N_p$ . Al igual que en el caso anterior cada simulación implica un despliegue distinto de la red (Se ejecuta *SimularRed* cada vez) y unas sesiones de comunicación entre los nodos cognitivos diferentes.

Para cada simulación los parámetros de entrada son:

Dimensiones de la red:  $XMAX = 20$ ,  $YMAX = 20$

Número de nodos cognitivos:  $N_s = 10$

Factores de actividad de los primarios:  $FA = 0,5$

Número de sesiones:  $Num\_Sesiones = 4$

Longitud de datos de cada sesión:  $LongDatos = 3000$

Potencia transmitida:  $Pot = 1$ ,  $PotPrim = 1$

Sensibilidad de los nodos:  $Sens = 0,01$ ,  $SensPrim = 1^{-3}$

Coefficiente de pérdidas de propagación:  $alfa = 2,5$

Quantum de tiempo:  $quantum = 0,1$ .

El comportamiento de la red con respecto al número de nodos primarios existentes se percibe de forma clara: conforme aumenta  $N_p$  aumentan los recursos que puede aprovechar la red cognitiva, ya que ésta puede utilizar un mayor número de bandas de frecuencia. Por tanto para un Factor de Actividad constante, si  $N_p$  es aumentado, mejoran todos los parámetros de la red: se reduce el retardo y aumentan el throughput y la tasa de ocupación de los enlaces. Además se observa como la mejora obtenida sigue una función exponencial, de forma que aumentar  $N_p$  cuando presenta un valor pequeño implica una mejora mucho mayor que hacerlo cuando  $N_p$  es elevado. Esto se debe a que cuando el número de frecuencias es muy elevado, el factor limitante en el comportamiento de la red ya no es el número de frecuencias disponibles, sino otros factores que dependen de la calidad de los enlaces, como la potencia transmitida, el ancho de banda, el coeficiente de pérdidas, etc.

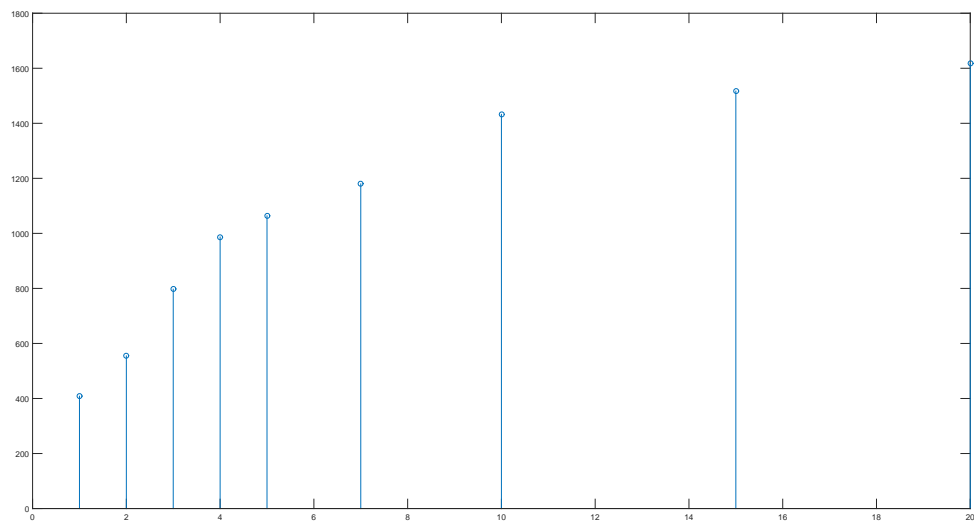


Figura 9: Througput Util en función del Número de nodos Primarios.

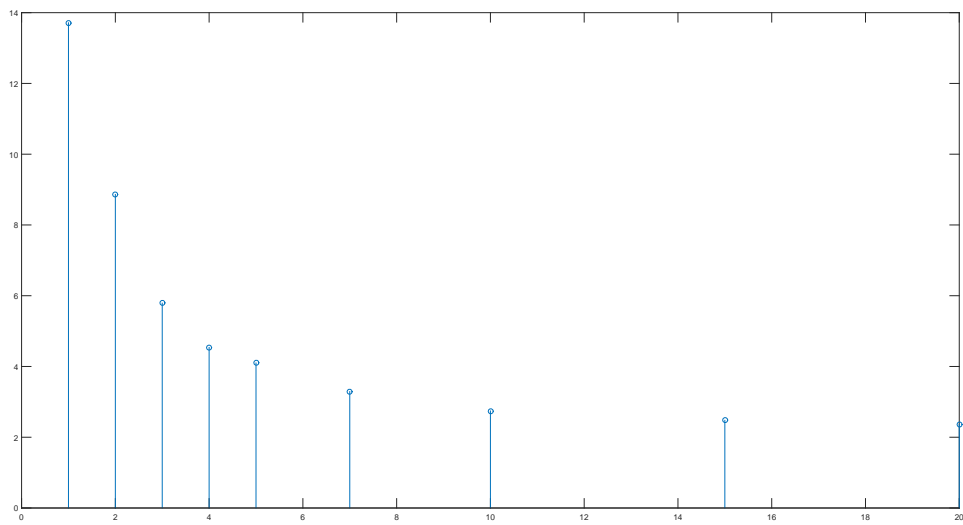


Figura 10: Retardo en función del Número de nodos Primarios.

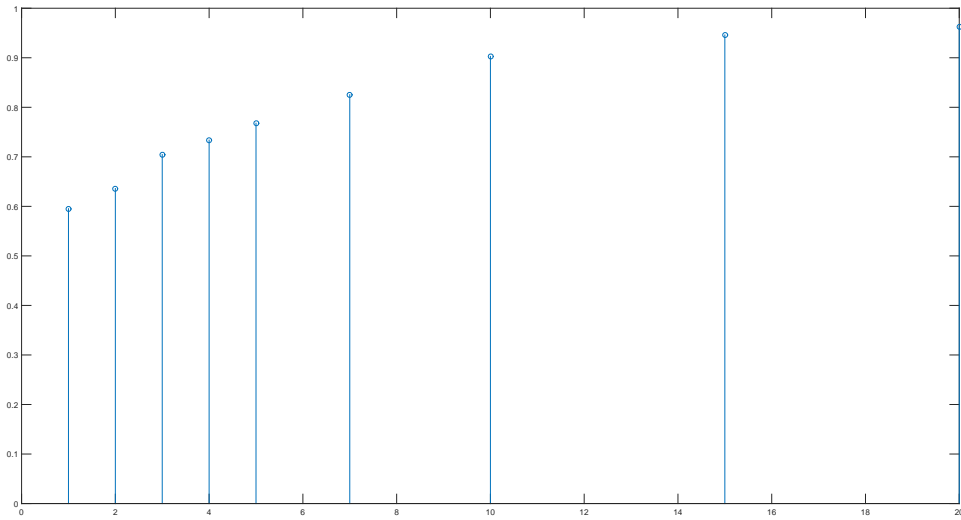


Figura 11: Tasa de ocupación de los enlaces en función del Número de nodos Primarios.

### 5.3. Influencia de FA:

El comportamiento de la red cognitiva con respecto a la actividad de los primarios se observa de forma clara. Cuando aumenta el valor de FA, es decir cuando los nodos primarios son más activos, la red cognitiva empeora sus prestaciones, lo cual resulta obvio, pues la red cognitiva se basa en aprovechar los recursos de la red primaria que ésta no utiliza. Si los nodos primarios tienen menos probabilidad de estar inactivos, entonces la red cognitiva tiene menos recursos disponibles. Además se aprecia como, cuando la tasa de actividad de los primarios presenta un valor bajo, el comportamiento es similar ( $FA = 0,1$  ó  $FA = 0,3$ ), debido a que la red no está limitada por éste parámetro. Sin embargo cuando el Factor de actividad de los nodos primarios es elevado, un pequeño cambio en el valor de FA se traduce en un cambio significativo en el comportamiento de la red, debido a que en éste caso el factor de actividad sí es un limitante de las prestaciones de la misma.

Se puede observar como aumentar el factor de actividad tiene un efecto similar a reducir el número de nodos primarios de la red. En ambos casos se están reduciendo los recursos de los que dispone la red cognitiva. Además, el cambio en el comportamiento de la red al variar estos parámetros es exponencial, de forma que cuando el parámetro presenta un valor muy malo ( $N_p$  muy pequeño o FA muy alto) una pequeña variación influye de forma significativa en la red. Sin embargo conforme el parámetro va tomando valores mejores (FA se va reduciendo y  $N_p$  va aumentando), el impacto que supone para la red una variación del parámetro se va reduciendo.



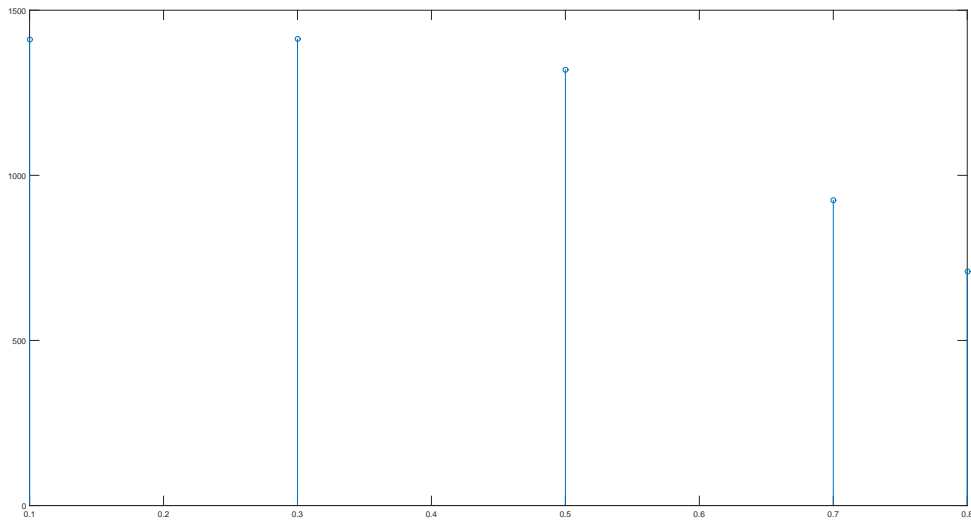


Figura 12: Througput Util en función del Factor de Actividad de los nodos primarios.

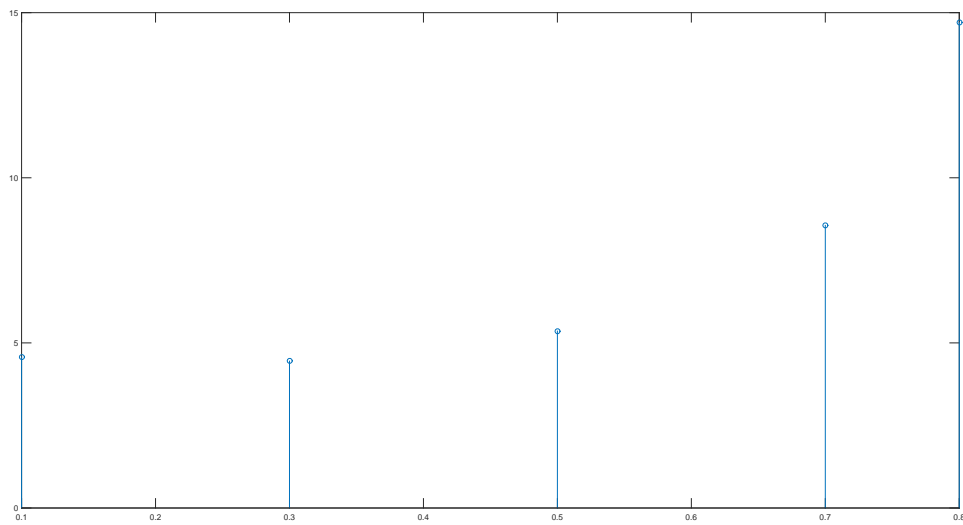


Figura 13: Retardo en función del Factor de Actividad de los nodos primarios.

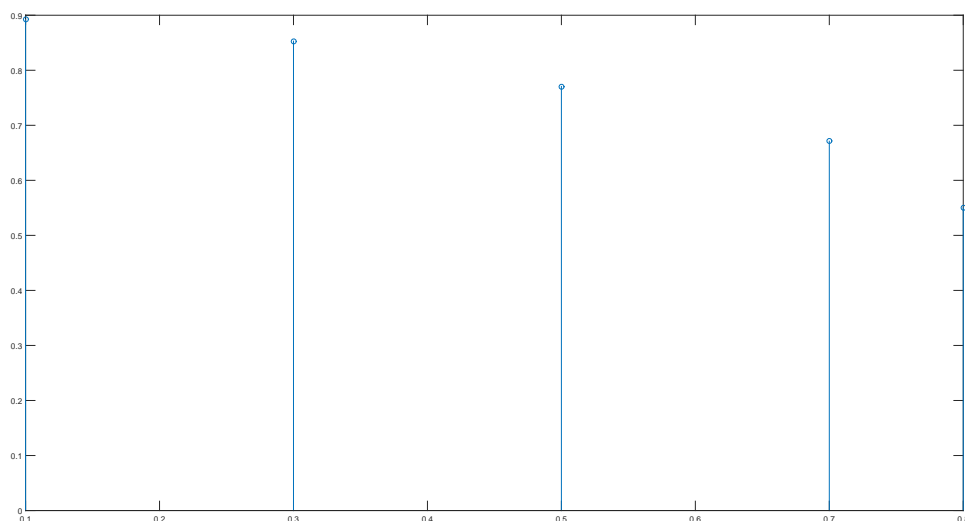


Figura 14: Tasa de ocupación de los enlaces en función del Factor de Actividad de los nodos primarios.

## 6. Conclusiones y Posibles ampliaciones del simulador

En este documento se ha introducido y explicado el simulador de red cognitiva implementado. Para ello, en primer lugar se ha hecho una breve introducción al tema de las redes cognitivas para posteriormente explicar los algoritmos de encaminamiento implementados y la estructura del simulador. Posteriormente se ha presentado un ejemplo de una simulación y se ha analizado cómo varía el comportamiento de la red en función de algunos parámetros.

En cuanto a los algoritmos de encaminamiento implementados, cabe destacar que el algoritmo GYMKHANA resulta muy ineficiente cuando el número de nodos de la red es elevado. Esto se debe a que se generan muchas operaciones, entre las que se incluyen propagar las *Route Requests* a lo largo de la red para hallar todos los caminos, evaluar cada una de las *Route Requests* generando un grafo virtual y calculando la matriz Laplaciana de dicho grafo virtual y generando una función de utilidad por cada camino posible, para luego escoger aquél camino que presenta la función de utilidad máxima. Posteriormente a la selección del camino, se deben generar las *Route Reply* para informarle a cada nodo la decisión.

Es posible que la ineficiencia del simulador frente a este algoritmo de encaminamiento tenga que ver con el hecho de que se implementa en *Matlab*, que, por que el autor ha investigado, no es muy eficiente en cuanto al uso de clases, sin embargo, al menos en la solución implementada resulta inviable para simulaciones con un gran número de nodos.

Por otra parte el algoritmo de retardo mínimo modificado presenta buenas prestaciones y permite simular redes con un número elevado de nodos y con cantidades de datos a transmitir por la red cognitiva también elevados. La implementación de este algoritmo carece de la simulación del canal de control común, que simularía las comunicaciones sobre

el estado de la red que llevan a cabo los nodos entre sí.

A continuación se van a comentar algunas posibles ampliaciones en la funcionalidad del simulador:

- Puede ampliarse el modelo de propagación del simulador para tener en cuenta el efecto del *fading*. De esta forma se tendría en cuenta no sólo las irregularidades del terreno a través del *shadowing*, sino que se considerarían los efectos de interferencias temporales y espaciales propios de las comunicaciones inalámbricas y que se modelan a través de un modelo estadístico que simule el canal con fading. Esto implicaría que la señal puede tener grandes variaciones con respecto a su nivel medio, por lo que podrían perderse paquetes cuando el canal presente un desvanecimiento. Podría ser útil implementar algún mecanismo que cuantifique los efectos del nivel de fading sobre el comportamiento total de la red. Además, podrían considerarse distintos modelos de fading (Como Rayleigh o Rice) que se corresponden con diferentes situaciones físicas, para comparar los comportamientos entre unos y otros.
- En el simulador implementado, el modo de operación de los nodos cognitivos es *on/off*, esto quiere decir que cuando la interferencia que generan en el primario es menor que un valor umbral, el nodo cognitivo transmite a una potencia fija y cuando genera una interferencia mayor no transmite. Este esquema podría modificarse por otro en el cual el nodo cognitivo adapte su potencia de transmisión al estado que presenta el canal nodo cognitivo - nodo primario, de forma que la potencia recibida siempre esté por debajo del valor umbral permitido. Esto supone que el nodo cognitivo debe tener conocimiento del estado del canal en cada instante, pero si se tiene en cuenta que el nodo cognitivo debe monitorizar el estado del primario de todas maneras, no es descabellado de realizar. Siguiendo este esquema la capacidad del enlace no es fija sino que depende del estado del canal, teniendo en algunas ocasiones un canal de pobres prestaciones, pero permitiendo la transmisión mientras que en el esquema implementado estaría prohibida. Además, cuando el canal nodo cognitivo-nodo primario es de muy mala calidad, el nodo cognitivo podría aprovecharlo para transmitir a una potencia mayor y obtener una mejor SNR.
- En una red cognitiva los nodos cognitivos compiten entre sí por los recursos que los nodos primarios dejan disponibles. En este simulador, cuando dos nodos cognitivos compiten por un mismo recurso se asigna a uno de ellos de forma aleatoria y éste no lo desaloja hasta que ha finalizado su transmisión o hasta que lo expulsa el primario. Evidentemente este es un método de resolución de contiendas muy básico y en el simulador podría implementarse algún mecanismo más elaborado en el que se tengan en cuenta más factores como el tiempo que lleva esperando el nodo o el tiempo que lleva ocupando el medio. Por lo tanto una clara ampliación del simulador sería implementar métodos de Control de Acceso al Medio (MAC) más eficientes.
- En el simulador se indican las comunicaciones de la red cognitiva como parámetro de entrada, a través del número de sesiones a simular. Esto también podría modificarse, generando un parámetro similar al Factor de Actividad de los nodos primarios en los nodos cognitivos. De esta forma se indicaría el tiempo a simular, en lugar del número de sesiones cognitivas. En función de la actividad de los nodos cognitivos, se generarían una serie de sesiones durante el tiempo a simular.

- Otra ampliación que se podría realizar es la de añadir la simulación del canal de control común en el simulador cuando se utiliza el algoritmo de menor peso modificado. Esto incluye que los nodos cognitivos lleven a cabo algún mecanismo de estimación de la actividad de los nodos primarios para luego transmitir los datos por el canal de control común. D

## Referencias

- [1] A. Abbagnale and F. Cuomo. Gymkhana: A connectivity-based routing scheme for cognitive radio ad hoc networks. In *Proc. INFOCOM IEEE Conf. Computer Communications Workshops*, pages 1–5, March 2010.
- [2] A. Jamakovic and S. Uhlig. On the relationship between the algebraic connectivity and graph’s robustness to node and link failures. In *Proc. rd EuroNGI Conf. Next Generation Internet Networks*, pages 96–102, May 2007.
- [3] L. Yun, Q. Fengxie, L. Zhanjun, and Z. Hongcheng. Cognitive radio routing algorithm based on the smallest transmission delay. In *Proc. 2nd Int Future Computer and Communication (ICFCC) Conf*, volume 2, pages V2–306–V2–310, May 2010.