

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y Electrónica



TRABAJO DE DIPLOMA

**Análisis de estrategias de ciclo de trabajo de radio
en Redes de Sensores Inalámbricos.**

Autor: Marlin de la Caridad Castro Castro

Tutor: Dr.C. Vitalio Alfonso Reguera

Santa Clara

2015

"Año 57 del Triunfo de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Telecomunicaciones y electrónica



TRABAJO DE DIPLOMA

Análisis de estrategias de ciclo de trabajo de radio en Redes de Sensores Inalámbricos.

Autor: Marlin de la Caridad Castro Castro

mccastro@uclv.edu.cu

Tutor: Dr.C. Vitalio Alfonso Reguera

Profesor titular, departamento de Telecomunicaciones y
Electrónica, Facultad Ingeniería Eléctrica,
vitalio@uclv.edu.cu

Santa Clara

2015

"Año 57 del Triunfo de la Revolución"



Hago constar que el presente trabajo de diploma fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de estudios de la especialidad de Ingeniería en Telecomunicaciones y Electrónica, autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma del Jefe de Departamento
donde se defiende el trabajo

Firma del Responsable de
Información Científico-Técnica

PENSAMIENTO

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber”.

Albert Einstein.

DEDICATORIA

De manera especial dedico esta tesis a mis padres, por la ayuda incondicional prestada en la creación de la misma, por la confianza que han depositado en mí a lo largo de mi vida, corrigiendo mis faltas y celebrando mis triunfos y por haberme inculcado la fortaleza para no desvanecerme en las peores situaciones, sino poder levantarme y aprender de ellas.

Además está dedicada a todos los interesados en esta línea de investigación para el desarrollo de futuras aplicaciones.

AGRADECIMIENTOS

Ofrezco mis agradecimientos a todos aquellos que de una forma u otra han contribuido a la realización de este trabajo de diploma, en especial a mi tutor y a mis padres.

TAREA TÉCNICA

- Caracterización de los principales mecanismos propuestos para el empleo de ciclos de trabajo de radio.
- Identificación de las medidas de desempeño fundamentales de los protocolos MAC para las RSI.
- Implementación con ayuda de un simulador de eventos discreto de los ciclo de trabajo de radio de los protocolos B-MAC, X-MAC y RI-MAC.
- Comparación los ciclo de trabajo de radio con relación a las siguientes medidas de desempeño: consumo energético, latencia y tasa de transmisión.

Firma del Autor

Firma del Tutor

RESUMEN

Las Redes de Sensores Inalámbricos necesitan de un tiempo de vida máximo, por lo que se requiere de un consumo energético mínimo. Para esto se establecen los ciclos de trabajo de radio en cada nodo sensor a partir de los protocolos control de acceso al medio (MAC). De todas las clasificaciones de ciclo de trabajo de radio, los de tipo asíncrono logran el mayor ahorro energético, pues tienen menor ciclo de trabajo que los tipos síncronos y semisíncronos. Dentro de los ciclos de trabajo asíncronos encontramos el muestreo de preámbulo largo, el muestreo de preámbulo corto y el iniciado en el receptor, los cuales son implementados en los protocolos B-MAC, X-MAC y RI-MAC respectivamente. Con ayuda de la herramienta de software MATLAB en este trabajo se implementaron estos tres mecanismos de ciclo de trabajo de radio. El desempeño de los mismos fue analizado variando el número de nodos, el tráfico y el ciclo de trabajo en una red simulada. A partir de los experimentos se demuestra que entre estos protocolos, el de mayor consumo energético y latencia es B-MAC, el mejor comportamiento en cargas de tráfico ligeras lo tiene X-MAC, y la mayor eficiencia y efectividad la tiene RI-MAC en cuanto a consumo energético y latencia en la transmisión de paquetes.

TABLA DE CONTENIDOS

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTOS	iii
TAREA TÉCNICA	iv
RESUMEN	v
INTRODUCCIÓN	10
CAPÍTULO 1. Clasificación de los Mecanismos de Ciclo de Trabajo de Radio en Redes de Sensores Inalámbricos.....	13
1.1 Ventajas y limitaciones del ciclo de trabajo de radio	13
1.2 Medidas de desempeño de los Mecanismos de Ciclo de Trabajo de Radio.	15
1.2.1 Consumo energético.	15
1.2.1.1 Consumo energético del hardware del nodo sensor.....	15
1.2.1.2 Consumo energético en la transmisión.	17
1.2.2 Latencia en la transmisión de paquetes.....	19
1.2.3 Tasa de Transmisión.	19
1.3 Clasificación general de los Mecanismos de Ciclo de Trabajo de Radio.	20
1.3.1 Mecanismos sincrónicos de Ciclo de Trabajo de Radio.	21
1.3.2 Mecanismos Semisíncronos de Ciclo de Trabajo de Radio.....	22
1.3.3 Mecanismos Asincrónicos del Ciclo de Trabajo de Radio.	24

1.4 Conclusiones del capítulo	28
CAPÍTULO 2. Implementación de los mecanismos de ciclos de trabajo de radio.	30
2.1 Ventajas de los ciclos de trabajo de radio asíncronos.....	30
2.2 Protocolos MAC que implementan los Ciclos de Trabajo de Radio Asíncrónicos.	31
2.2.1 Muestreo de preámbulo largo: B-MAC	31
2.2.2 Muestreo de preámbulo corto: X-MAC.....	32
2.2.3 Iniciado por el receptor: RI-MAC	32
2.3 Diseño utilizando la herramienta de software MATLAB de los mecanismos de ciclo de trabajo de radio.....	33
2.3.1 Herramienta de Software MATLAB	33
2.3.2 Diseño del Mecanismo de Ciclo de Trabajo de Radio: Muestro de preámbulo largo.	34
2.3.3 Diseño del Mecanismo de Ciclo de Trabajo de Radio: Muestro de preámbulo corto.	35
2.3.4 Diseño del Mecanismo de Ciclo de Trabajo de Radio: Iniciado en el receptor...36	
2.4 Conclusiones del capítulo.	37
CAPÍTULO 3. Resultados y discusión acerca de los mecanismos de ciclo de trabajo de radio simulados. 38	
3.1 Descripción del escenario de simulación.....	38
3.2 Experimento variando el número de nodos	39
3.3 Experimento variando el tráfico	43
3.4 Experimento variando el ciclo de trabajo	45
3.5 Conclusiones del capítulo.	48
CONCLUSIONES Y RECOMENDACIONES	50
Conclusiones.....	50

Recomendaciones	51
REFERENCIAS BIBLIOGRÁFICAS	52
GLOSARIO	56
ANEXOS	58
Anexo I Función para la implementación del mecanismo de ciclo de trabajo de radio: muestreo de preámbulo largo.....	58
Anexo II Función para la implementación del mecanismo de ciclo de trabajo de radio: muestreo de preámbulo corto.	63
Anexo III Función para la implementación del mecanismo de ciclo de trabajo de radio: iniciado en el receptor.	68
Anexo IV Experimento para analizar el comportamiento del consumo energético en función de la variación del número de nodos en la red.	74
Anexo V Experimento para analizar el comportamiento la latencia en la transmisión de paquetes en función de la variación del número de nodos en la red.....	76
Anexo VI Experimento para analizar el comportamiento de la tasa de transmisión en función de la variación del número de nodos en la red.	78
Anexo VII Experimento para analizar el comportamiento del consumo energético en función de la variación del tráfico.	80
Anexo VIII Experimento para analizar el comportamiento de la latencia en la transmisión de paquetes en función de la variación del tráfico.	82
Anexo IX Experimento para analizar el comportamiento de la tasa de transmisión en función de la variación del tráfico.	84
Anexo X Experimento para analizar el comportamiento del consumo energético en función de la variación del ciclo de trabajo.	85
Anexo XI Experimento para analizar el comportamiento la latencia en la transmisión de paquetes en función de la variación del ciclo de trabajo.....	88

Anexo XII Experimento para analizar el comportamiento la tasa de transmisión de paquetes en función de la variación del ciclo de trabajo.90

INTRODUCCIÓN

El ciclo de trabajo de radio por mucho tiempo ha sido usado en una gran cantidad de dispositivos para ahorrar energía y extender el tiempo de vida de los mismos. La aplicación del ciclo de trabajo de radio en los dispositivos inalámbricos de comunicación fue desarrollada en paralelo con la popularización de la telefonía móvil y la computación móvil, ya que tales dispositivos presentan baterías con poca energía, además su interfaz de radio que es raramente usada de forma continua, aporta un gran consumo energético. [1], [2].

Con el fin de ahorrar energía a partir del módulo de comunicaciones, se emplea el “ciclo de trabajo de radio”, que consiste en que el transceptor se despierte o se duerma en función de si el sensor espera recibir o no algún paquete. Para que esto sea posible, debe haber un protocolo de control de acceso al medio que haga que los nodos sensores se comuniquen entre sí, ya sea de manera síncrona o no. Con un protocolo síncrono se logra la disminución del consumo energético, la escucha ociosa, las colisiones y la sobre-escucha, a costa de aumentar el sobre-encabezado de control. Por su parte, los protocolos con ciclo de trabajo asincrónico logran además disminuir el sobre-encabezado de control, y como consecuencia logran ahorrar una mayor cantidad de energía. Algunos de los mecanismos que este ciclo utiliza son, el muestreo de preámbulo largo, el muestreo de preámbulo corto, el iniciado en el receptor, entre otros.

El ciclo de trabajo de radio es un mecanismo fundamental en las Redes de Sensores Inalámbricos (RSI), ya que se requiere que los nodos mantengan su funcionamiento por un largo tiempo antes de recargar la batería. Los nodos sensores necesitan ser pequeños y baratos, mientras que para la transmisión de datos, ellos tienen que transmitir algunos bytes por segundo o incluso al día. Esta combinación demanda baterías y ciclos de trabajo de radio pequeños.

El ciclo de trabajo puede ser lo más importante de todas las técnicas “verdes”. Es la más efectiva [2] y puede ser implementada en conjunto con otras técnicas como el control de potencia o enrutamiento. Es también ampliamente implementado en los principales sistemas operativos empleados en las RSI como, TinyOS [3] y Contiki [4].

Muchas propuestas diferentes de mecanismos de ciclo de trabajo han emergido durante la última década, lo cual evidencia la necesidad actual de lograr un ciclo de trabajo de radio efectivo. La forma usual de clasificarlos es conforme a la sincronización entre los nodos. Sin embargo, simplemente dividiendo las propuestas en síncrono y asíncrono, no es suficiente para proveer una buena comprensión sobre los ciclos de trabajo actuales, por lo que se adiciona a esta clasificación los mecanismos semisíncronos [5].

El objetivo fundamental que se persigue con los ciclos de trabajo es lograr el mínimo consumo energético, considerando también, la utilización del canal, y la disminución del retardo en las comunicaciones. En este sentido resulta conveniente conocer dentro de las estrategias actuales cuales presentan un mejor desempeño ante cambios en sus parámetros y variaciones de las condiciones de red tales como: el aumento del número de nodos, el incremento de tráfico, y la modificación del propio ciclo de trabajo.

El objetivo general que se persigue con este trabajo de diploma es:

- Evaluar diferentes estrategias para la implementación del ciclo de trabajo de radio utilizado por los protocolos MAC en las redes de sensores inalámbricos.

Además se establecen los siguientes objetivos específicos:

- Caracterizar los principales mecanismos propuestos para el empleo de ciclos de trabajo de radio.
- Identificar las medidas de desempeño fundamentales de los mecanismos de ciclo de trabajo para las RSI.
- Implementar con ayuda de un simulador de eventos discreto algunos de los protocolos de ciclo de trabajo de radio más usados.
- Comparar los mecanismos de ciclo de trabajo de radio seleccionados en cuanto a las principales medidas de desempeño para las RSI.

Con este trabajo se pretende contribuir al desarrollo de las Redes de Sensores Inalámbricos, a partir del estudio de los mecanismos de ciclos de trabajo de radio.

Además, se realiza una contribución al análisis del diseño de los ciclos de trabajo de radio y su relación con la reducción del gasto de energía en aras de lograr la mayor adaptación posible a los cambios en la red.

Se pretende también brindar elementos didácticos y métodos que permitan emprender investigaciones más profundas acerca de los nuevos protocolos MAC, teniendo en cuenta las aplicaciones más favorables para el país. Así como continuar desarrollando nuevos estudios, no solo sobre los mecanismos de ciclo de trabajo para la capa MAC, sino también sobre los demás protocolos que forman parte de la pila (stack) de protocolos de la red de sensores inalámbricos.

En el Capítulo I se clasifican los mecanismos de ciclo de trabajo de radio en cuanto a sus requisitos de sincronización y se exponen las principales fortalezas y debilidades de los mismos. Se establecen las medidas de desempeño de los protocolos MAC que implementan los ciclos de trabajo de radio.

En el Capítulo II se implementan con ayuda de un simulador de eventos discreto los protocolos seleccionados a partir de su ciclo de trabajo de radio.

En el Capítulo III se exponen los resultados de la comparación de los principales protocolos en cuanto a las medidas de desempeño, según el mecanismo de ciclo de trabajo implementado en cada uno ellos. Estas medidas son: consumo energético, latencia punto a punto, y tasa de transmisión.

CAPÍTULO 1. Clasificación de los Mecanismos de Ciclo de Trabajo de Radio en Redes de Sensores Inalámbricos.

Dentro de la clasificación de los ciclos de trabajo de radio se encuentran los de tipo síncronos, semisíncronos y asíncronos [5]. En este capítulo se analizan los mecanismos de ciclo de trabajo de radio correspondientes a cada una de estas clasificaciones, enfatizando en los de tipo asíncrono pues son los que ahorran la mayor cantidad de energía, objetivo fundamental que se persigue con el uso de los ciclos de trabajo de radio.

Este capítulo presenta tres epígrafes. En el primer epígrafe se muestran las ventajas y desventajas del ciclo de trabajo, en cuanto al comportamiento de la red. En el segundo epígrafe se establecen las medidas de desempeño fundamentales de los mecanismos de ciclo de trabajo de radio implementados en los protocolos MAC, como el consumo energético, la latencia en la transmisión de paquetes, y la tasa de transmisión. Por último en el tercer epígrafe veremos las distintas clasificaciones de los mecanismos de ciclos de trabajo de radio.

1.1 Ventajas y limitaciones del ciclo de trabajo de radio

Como se dijo anteriormente el objetivo fundamental del ciclo de trabajo de radio es reducir el consumo de energía de los nodos sensores y aumentar como consecuencia, el tiempo de vida global de la red. Entre sus ventajas se evidencia que al no tener la radio activa todo el tiempo se logra un ahorro energético considerable.

Los ciclos de trabajo apuntan a reducir la escucha ociosa (tener el transceptor del radio esperando en vano por una trama). Esto se evidencia en la dificultad que tienen los ciclos de trabajo de custodiar la radio en actividad sólo cuando es necesario, pues en la mayoría de las aplicaciones, los nodos no saben de antemano cuando llegan los datos.

La escucha ociosa no es la única fuente de gasto de energía. La sobre-escucha (cuando un nodo gasta energía en escuchar tramas que no son de su interés), el sobre-encabezado de control de paquetes y las colisiones, también son causas del gasto innecesario de energía. Es importante identificarlas pues, al intentar reducir la escucha ociosa, el ciclo de trabajo de

radio puede aumentar las tasas de colisión y puede introducir más tráfico de control, estos efectos secundarios pueden aumentar el mismo consumo de energía que el ciclo de trabajo trata de disminuir.

Las limitantes del ciclo de trabajo se evidencia en los parámetros que se deben sacrificar a cambio de un consumo energético eficiente como:

La demora del mensaje punto a punto:

Para transmitir un dato desde un nodo utilizando un ciclo de trabajo dentro de una red multisalto, ocasionalmente tendrá que esperar al siguiente salto para despertar o entrar en estado de actividad. Éste es llamado la espera en estado de reposo o sueño y puede significativamente sumarse a la latencia punto a punto. Algunas aplicaciones no podrán hacer frente a eso. Las aplicaciones de vigilancia, por ejemplo, necesitan garantías de que un acontecimiento dado le será comunicado oportunamente. De hecho, para aplicaciones de tiempo reales, el problema no es sólo el retardo añadido sino también la manera desigual con que ese retraso ocurre (para más información el lector es referido a [6]).

Las tasas de colisión:

Otro efecto secundario del ciclo de trabajo de radio es el acortamiento del tiempo de transmisión y de recepción. Si un protocolo MAC basado en CSMA (del inglés, *Carrier Sense Multiple Access*) es usado, entonces estas ventanas de tiempo más pequeñas aumentarán la probabilidad de colisiones. Si está basado en TDMA (del inglés, *Time Division Multiple Access*), entonces se necesita de sincronización y no existirá tasa de colisión pero la sincronización traerá consigo un incremento en el tráfico de control y un derroche de energía en su transmisión.

Sobre-encabezado de control de paquete:

El ciclo de trabajo de radio puede necesitar tráfico adicional de control. La fuente más común de sobre-encabezado de control es la sincronización. La sincronización del ciclo de trabajo en la red requiere de una actualización frecuente del periodo de sincronismo para evitar los desplazamientos del reloj que se encuentran fuera del margen de error. En estos casos los diseñadores deben inspeccionar si el gasto de energía añadido a causa del tráfico adicional de control, es compensado por los ahorros del ciclo de trabajo de radio.

1.2 Medidas de desempeño de los Mecanismos de Ciclo de Trabajo de Radio.

Entre las medidas de desempeño fundamentales en las redes de sensores inalámbricos se evidencia el tiempo de vida útil de la misma, lo cual se manifiesta en el ciclo de trabajo como consumo energético. Por lo que, para un mayor tiempo de vida en la red se necesita un ciclo de trabajo de radio que garantice el mínimo de consumo de energía. Esto nos lleva a la principal medida de desempeño de los ciclos de trabajo de radio: el consumo energético. También en las RSI, es importante la demora dentro de la red para el logro de una alta calidad del servicio. Esto se evidencia en el ciclo de trabajo como la latencia en la transmisión de los paquetes, lo que nos lleva a otra importante medida de desempeño: la latencia punto a punto. Además las RSI requieren una tasa de transmisión de datos acorde con las necesidades para las que fue creada la red, de allí que tengamos otra importante medida en consideración: la tasa de transmisión.

1.2.1 Consumo energético.

Como los sensores generalmente son dispositivos alimentados con batería, los aspectos críticos a afrontar tienen que ver con intentar reducir el consumo energético de estos, para que el tiempo de vida de la red pueda ser extendido a tiempos razonables y así lograr un mayor aprovechamiento de los servicios que ofrece la misma.

El consumo energético de los sensores depende en gran medida del ciclo de trabajo de radio que utilicen, además de las características propias del nodo como el consumo del hardware y la energía utilizada en la transmisión y recepción de los datos.

1.2.1.1 Consumo energético del hardware del nodo sensor.

En esta sección se describen los principales bloques funcionales del nodo sensor. Estos bloques producen pérdidas energéticas de manera individual, estas pérdidas serán en mayor o menor medida dependiendo de la función que realizan los mismos. También se discuten las principales fuentes de pérdida energética. Finalmente se muestran los valores de consumo de energía de algunos sensores.

Un nodo sensor está compuesto de 4 partes fundamentales, estas son: unidad energética, la cual contiene a la batería y un número de reguladores de voltaje, una Unidad de Procesamiento que lo constituyen una memoria y un pequeño procesador, los Sensores

Físicos y el Circuito Transceptor, el cual lo constituye un Sistema de Radio, encargado de la transmisión y de la recepción. La Figura 1.1 muestra el diagrama en bloque de un nodo sensor.

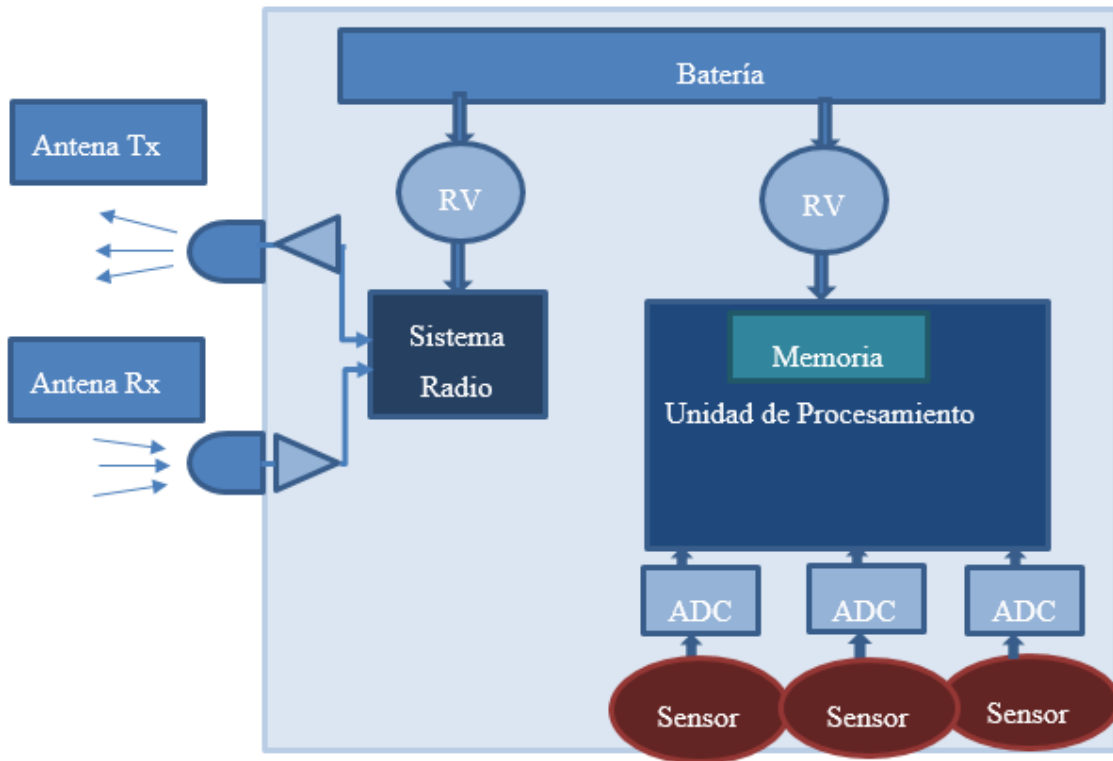


Figura 1.1: Diagrama en bloque de un nodo sensor inalámbrico.

La Unidad de Procesamiento es la encargada de leer los sensores físicos, extraer la información relevante desde los datos digitalizados e implementar los protocolos de red. Además determina la energía y la capacidad de cómputo del nodo.

El Sistema de Radio permite la comunicación inalámbrica entre los nodos de la red y el medio exterior. Los factores como el esquema de modulación, la razón de transmisión de los datos en la red, la potencia de transmisión, y el ciclo de trabajo de radio pueden afectar directamente las características energéticas de la red. Los nodos trabajan generalmente en tres modos de operación básicos: activo cuando el nodo se encuentra transmitiendo o recibiendo, escucha ociosa cuando está escuchando el canal sin establecer ninguna operación, y en estado de sueño cuando el nodo se encuentra en reposo. Contrario a la

creencia popular, los estudios sobre las RSI y los protocolos de ruteo afirman que la energía consumida en la escucha ociosa es alta, comparable con la energía consumida en el modo activo [7]. Es por esta razón que se recomienda un apagado completo del radio transceptor cuando no está en uso. En el Sistema de Radio se deben considerar varios acontecimientos que traen consigo el incremento del consumo energético como un cambio del estado del sistema y los efectos transitorios del transceptor.

Los sensores transforman una magnitud física o química en una señal eléctrica. Algunos ejemplos de estas magnitudes son la temperatura, la distancia, la aceleración, la inclinación, el desplazamiento, la presión, la humedad y el pH. La señal eléctrica puede ser creada a partir de un cambio producido en el sensor como un cambio de resistencia o capacidad.

Las baterías son elementos de gran complejidad, su buen trabajo depende de muchos factores, como su tamaño, el material del electrodo, y la razón de difusión del material activo en el electrolito. Existen diversos tipos de baterías ya sea recargables o no, que son usadas en las RSI. Según su composición electromecánica, la cual determina la eficiencia energética, pueden distinguirse baterías de Ni-Cd, Ni-Zn, Ag-Zn, Ni-MH y baterías de ion de litio, entre muchas otras. Los efectos importantes a considerar dentro de las baterías son el efecto de relajación y el efecto de la capacidad nominal [8], los cuales a largo plazo determinan el tiempo de vida de la batería.

Los reguladores de voltaje son los encargados de proveer los voltajes necesarios para los circuitos en el nodo. Los reguladores de voltaje lineales provocan grandes pérdidas energéticas, aproximadamente la corriente de salida multiplicada por la caída de voltaje a través del dispositivo, comparados con los reguladores conmutados, que aportan una pérdida energética en el rango de 5-15%. En cuanto al consumo de energía el diseño del área de los reguladores de voltaje tiene un gran impacto.

Es posible recargar las baterías con la energía extraída del medio como la luz, el viento, la vibración [9] y los campos electromagnéticos.

1.2.1.2 Consumo energético en la transmisión.

Tanto la transmisión como la recepción conllevan a circuitos electrónicos de gran consumo energético. La distancia entre el transmisor y el nodo receptor debe ser tomada en cuenta en el cálculo de la potencia disipada [10]. La Figura 2.2 muestra el diagrama en bloque de la

transmisión-recepción de un nodo a otro en las RSI. La Ecuación 1.1 describe como estimar la energía consumida por bit, en el modelo representado en la Figura 1.2 [11].

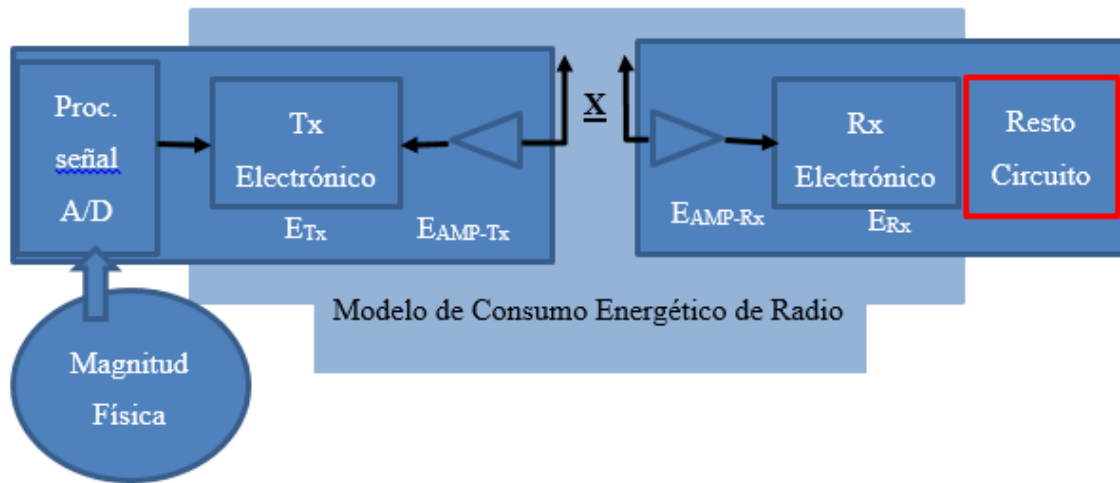


Figura 1.2: Modelo de Consumo Energético de Radio.

$$\text{Consumo energético} = P_{tx} * T_{tx} + P_{rx} * T_{rx} + P_{es} * T_{es} + P_d * T_d \quad (1.1)$$

Donde:

Consumo energético: Energía consumida por nodo.

P_{tx}: Potencia consumida en la transmisión.

T_{tx}: Tiempo que se mantiene transmitiendo.

P_{rx}: Potencia consumida en la recepción.

T_{rx}: Tiempo que se mantiene recibiendo.

P_{es}: Potencia consumida en la escucha.

T_{es}: Tiempo que se mantiene escuchando.

P_d: Potencia consumida en el periodo de sueño.

T_d: Tiempo que se mantiene dormido.

Una primera medida del consumo energético consiste, en ajustar la potencia de transmisión a las características de la trayectoria de propagación, como la atenuación y el radio de acción. Una aproximación usualmente utilizada para controlar la actividad del nodo, es conmutando el modo de operación entre los modos activo, de escucha ociosa y de reposo.

En el modo de sueño, el dispositivo consume la menor energía ya que en este estado el transmisor es apagado, y la frecuencia del procesador es reducida, lo que imposibilita el procesamiento de datos.

Una gran cantidad de tiempo es requerida para entrar y salir de estos modos. El estado intermedio entre el modo activo y el de sueño es la escucha ociosa, en la cual el nodo consume menos energía que en el modo activo ya que no realiza el procesamiento de datos.

1.2.2 Latencia en la transmisión de paquetes.

La latencia en la transmisión de los paquetes depende en gran medida del ciclo de trabajo, pues existen varios mecanismos que establecen demoras diferentes, debido a la estrategia de ahorro energético que utilicen. Es por esto que se deben implementar los ciclos de trabajo de radio en correspondencia con la aplicación en la cual tienen lugar.

Como establecimos antes, la latencia en la transmisión de paquetes en las RSI depende de la aplicación implementada en la misma. Por ejemplo en aplicaciones como la vigilancia, cuando se detecta un evento se necesita enviarlo a la estación base lo más pronto posible, para que la medida a llevar a cabo se tome con rapidez, en este caso se utiliza un protocolo síncrono, que aunque no tiene un buen rendimiento energético en comparación con las demás clasificaciones, tiene la latencia necesaria para transmitir los datos con rapidez.

En [12] se establece la estimación de la distribución de la latencia punto a punto. Su objetivo es calcular la probabilidad de que una secuencia de mensajes, pueda ser transmitida a través de una trayectoria dada, dentro de un espacio definido de tiempo. A partir de esto es posible realizar la estimación de tal distribución con una precisión favorable y baja complejidad.

Para una red multisalto la distribución de la latencia punto a punto se hace analizando la demora del reenvío de paquetes en cada salto intermedio y luego con la composición de una distribución punto a punto aproximada, se establece la latencia en la trayectoria.

1.2.3 Tasa de Transmisión.

La tasa de transmisión se refiere a la razón promedio de paquetes recibidos exitosamente por unidad de tiempo (generalmente medida en bits por segundos), la forma de calcularla es la siguiente:

$$Tasa\ de\ transmisión = \frac{Paquetes\ transmitidos}{Tiempo\ Total} \quad (1.2)$$

El límite para el rendimiento de la tasa de transmisión de una red es definido como la tasa de transmisión estable máxima de la red [13]. La máxima tasa de transmisión estable, es la máxima cantidad de tráfico por unidad de tiempo que puede ser observado en las RSI, mientras el tamaño de cola en cualquier nodo de la red sea soportable. En la literatura se refieren al límite del rendimiento de la tasa de transmisión con el término de capacidad, para definir la máxima tasa de transmisión alcanzable.

1.3 Clasificación general de los Mecanismos de Ciclo de Trabajo de Radio.

Los ciclos de trabajo están usualmente clasificados en sincrónicos y asincrónicos en relación al mecanismo usado para coordinar los períodos de actividad de los nodos sensores. Los métodos sincrónicos son esos donde los nodos están sincronizados en el tiempo, aunque el grado de sincronización varíe grandemente. En los métodos asincrónicos, los nodos no están obligados a conservar un reloj común.

Como la implementación en conjunto de los métodos de sincronización sincrónicos y asíncronos es más fácil de lograr que la sincronización global [14], tiene sentido agrupar a los nodos vecinos en clústeres sincronizados y que estos interactúen entre sí de manera asincrónica, resultando en un diseño semisíncrono (también llamado híbrido o semiasincrónico). Las diferentes categorías de clasificación están resumidas en la Figura 1.3 y se discute en los siguientes epígrafes.

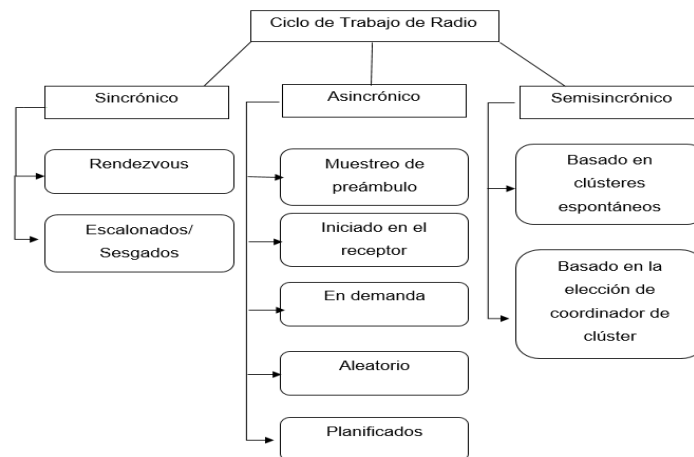


Figura 1.3: Clasificación de los mecanismos de Ciclo de Trabajo de Radio.

1.3.1 Mecanismos sincrónicos de Ciclo de Trabajo de Radio.

En los ciclos de trabajo síncronos, los nodos conservan referencias comunes de tiempo. Esto necesariamente no tiene que ajustarse con un tiempo global, pero si trae consigo el intercambio de información de sincronización, con el objetivo de lograr y conservar el grado necesario de sincronización a todo lo largo de la red. En esta clasificación, los ciclos de trabajo sincrónicos estarán dentro de dos mecanismos: el Rendezvous (estrictamente sincrónico) y el Escalonado/Sesgado. En la Figura 1.4 se resume la categoría de ciclo de trabajo sincrónico.

1. Rendezvous o modelo estrictamente sincrónico:

Rendezvous es un mecanismo de sincronismo global, donde todos los nodos activarán y apagarán sus radios al mismo tiempo. En un medio multisalto, el sincronismo global es difícil de lograr, ya que los errores de sincronismo tienden a acumularse. Por lo que este mecanismo debe hacer frente con un cierto grado de error de sincronización. Para conservar el grado de error tolerable, es necesario el uso de protocolos de sincronización y ocasionalmente de hardware adicional. Además, es usualmente necesario el intervalo de guarda, o sea, los nodos encenderán su radio ligeramente antes del tiempo establecido o de referencia.

El mecanismo Rendezvous trabaja muy bien en el tráfico de radiodifusión. Si se logra la sincronización global, entonces los diseños basados en Rendezvous pueden ser ventajosos, porque de su alta capacidad para coordinar las transmisiones, se reduce la escucha ociosa y las colisiones. Sin embargo, la sincronización global es muy difícil de lograr, por lo que es necesario el uso de hardware adicional y/o cantidades significantes de mensajes de control.

2. Escalonados o Sesgados:

Los modelos Rendezvous, no son la solución óptima en términos de minimizar la demora punto a punto. Ha sido observado que cuando una trama pasa a través de la red multisalto, desde un nodo sensor hasta la estación base, puede sufrir el problema de interrupción en el reenvío de datos. Esto ocurre cuando el flujo de datos se entrecorta por un nodo dentro de la trayectoria ascendente, que se encuentra en estado de reposo, pues el nodo sin saber que la trama se ha transmitido, se va a estado de sueño antes de que la misma llegue a él.

Para solucionar el problema de interrupción de reenvío de datos y reducir la espera de sueño, es propuesto el mecanismo de ciclo de trabajo escalonado o sesgado. En los protocolos MAC: D-MAC (del inglés, *Dynamic MAC*) [15], PELLMAC (de inglés, *Power efficient and low latency MAC*) [16], LEEMAC (del inglés, *Latency and energy efficient MAC*) [17], SPEED-MAC (del inglés, *Speedy and energy efficient data delivery MAC protocol*) [18], y CUPID (del inglés, *Communication pattern informed duty cycling*) [19] una topología de árbol es formada, originada en la estación base, y los periodos del ciclo de trabajo de los nodos seguirán un patrón de escalera, según la topología de árbol establecida.

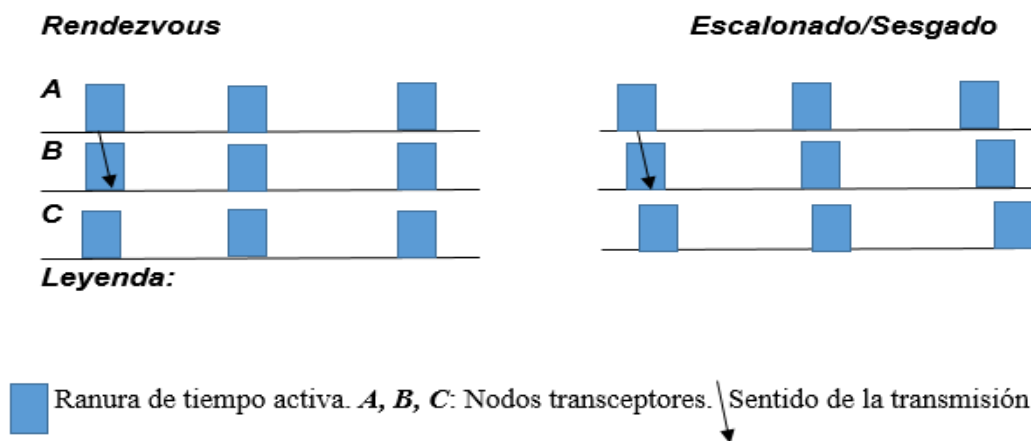


Figura 1.4: Ciclos de Trabajo de Radio Sincrónico.

1.3.2 Mecanismos Semisíncronos de Ciclo de Trabajo de Radio.

En las propuestas semisíncronos, los vecinos son agrupados sincrónicamente en clústeres y los clústeres interactúan entre sí asincrónicamente. Estos mecanismos toman lo mejor de los modelos sincrónicos y asincrónicos. La ventaja principal de la sincronización dentro de los clústeres, es que esta es más fácil de lograr entre vecinos que globalmente dentro de la red. El clúster puede utilizar mecanismos de elección, requiriendo de tráfico de control, lo cual puede ser inadecuado para topologías dinámicas. Sin embargo, hay también mecanismos que formarán los clústeres espontáneamente a consecuencia de la sincronización entre vecinos.

A continuación se describen las principales características de los mecanismos semisíncronos, los cuales están enmarcados en dos mecanismos fundamentales: el

agrupamiento espontáneo y el basado en la elección de un nodo cabecera, los cuales se muestran en la Figura 1.5.

1. Agrupamiento espontáneo:

El agrupamiento espontáneo en clústeres, se refiere al mecanismo donde los nodos se coordinan por ellos mismos, sin la necesidad de un nodo principal en cada grupo. Tienen la desventaja de que un nodo intermedio entre dos clústeres adoptará los ciclos de trabajo de los nodos vecinos, estos perteneciendo por separado a uno de los dos clústeres, resultando que el nodo toma los ciclos de trabajo de ambos clústeres, lo que aumenta el consumo de energía.

2. Elección del Nodo-Cabecera en el clúster:

En el agrupamiento de los nodos dentro del clúster basado en la elección de un nodo coordinador, uno de los nodos (el principal) recibe la asignación especial (temporalmente, en la mayoría de los casos) de actividad coordinativa del grupo.

Los problemas que este mecanismo debe resolver incluyen encontrar una elección eficiente del nodo cabecera del clúster, y los mecanismos de coordinación inter-clúster, así como formas eficientes para la retransmisión de datos inter-clúster. En términos de ciclo de trabajo, los protocolos que utilizan el mecanismo de ciclo de trabajo basado en la elección de cabecera de clúster tienden a adoptar una forma de TDMA para comunicaciones inter-clúster, lo que sugiere el uso del mecanismo Rendezvous en el que los nodos sincronizan sus ciclos de trabajo con el de su cabecera de clúster. Sin embargo, la comunicación inter-clúster es difícil de lograr con esquemas síncronos y a causa de esto, por lo general, se utiliza cabeceras de clústeres que operan con ciclos de trabajo más altos, ya que en la mayoría de los casos, también son responsables de la retransmisión inter-clúster.

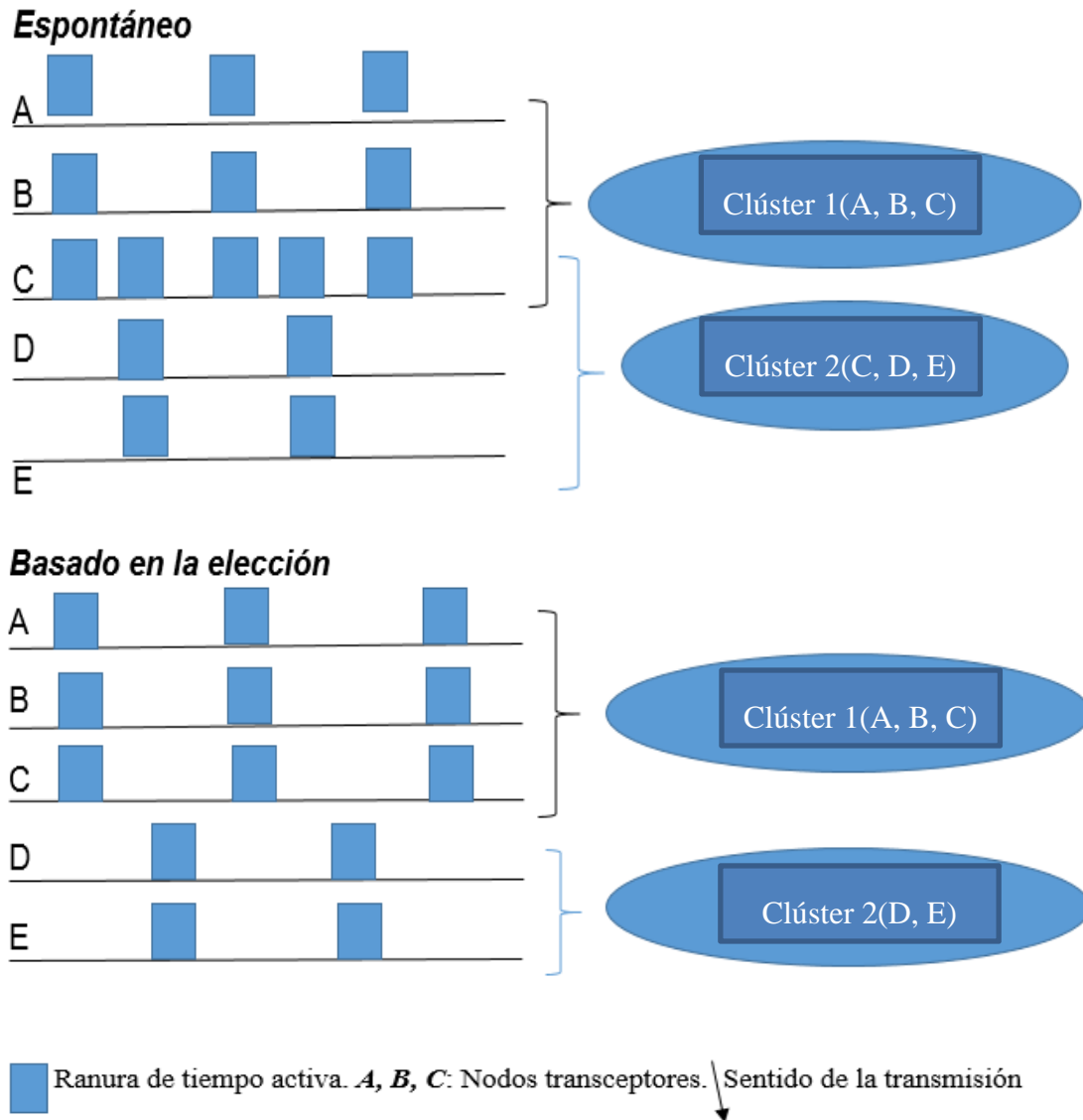


Figura 1.5: Ciclos de Trabajo de Radio Semisincrónico.

1.3.3 Mecanismos Asincrónicos del Ciclo de Trabajo de Radio.

La sincronización de los nodos de una RSI multisalvo es difícil y costosa, que requiere capacidades de hardware o de procesamiento adicionales. Además, la adición frecuente de tráfico de control trae consigo demora de propagación y gastos de energía en la transmisión [14]. Como respuesta a esto se propuso los mecanismos de ciclos de trabajo asíncronos (en la que los nodos no tendrán que ponerse de acuerdo sobre las referencias del tiempo).

A continuación se describen los mecanismos de los ciclos de trabajo asíncronos. Los mismos se ilustran en la Figura 1.6.

1. Muestreo de preámbulo largo:

Una técnica importante, incorporada en RSI en 2004 por B-MAC (del inglés, *Berkeley Medium Access Control*) [20] y WiseMAC [21], es el muestreo de preámbulo, a veces referido como LPL (del inglés, *Low Power Listening*) en el contexto de las redes de sensores. La idea es reducir la escucha ociosa mediante la transferencia del gasto de energía al transmisor (sólo uno) y retirarlo del receptor (posiblemente muchos). El funcionamiento de este mecanismo se basa en que los nodos se van a dormir de forma asíncrona y se despiertan periódicamente para comprobar la actividad del canal. Las tramas están precedidas por un largo preámbulo, de longitud mayor que la duración de los tiempos activo y de reposo juntos; con el objetivo de que cualquier nodo tenga tiempo de despertar, detectar la transmisión de preámbulo, y permanecer activo para recibir la trama entrante, si es necesario.

Este mecanismo es muy utilizado en el sistema operativo TinyOS, pero sus desventajas hacen que no se utilice en gran medida. Entre estas desventajas podemos nombrar la larga apropiación del canal por el preámbulo que no sólo será un desperdicio de energía, sino que también evita la posibilidad de que otros nodos puedan transmitir. Además la latencia de extremo a extremo puede ser demasiado grande y, por último, existe una sobre-escucha excesiva, puesto que los nodos que no están involucrados en la comunicación también permanecen activos mientras se escucha el preámbulo. TinyOS apoya LPL sólo para los radios CC2420 y CC1000 y no es actualmente compilado por defecto debido a la utilización de memoria adicional [22].

2. Muestreo de preámbulo corto

El sistema asíncrono se ha mejorado con la introducción de técnicas de preámbulo corto, como en X-MAC [23], que utiliza en lugar del largo preámbulo, un tren de tramas cortas intermitentes. Las tramas cortas, con cortos intervalos entre ellas, tienen la ventaja de ser interrumpibles, dando al receptor la oportunidad de reconocer la intención del transmisor y disminuir el tiempo de señalización. El receptor envía un beacon para indicarle al transmisor que puede enviar los datos. Además, en lugar de enviar un preámbulo largo sin información utilizable, estas tramas cortas pueden llevar la dirección del nodo receptor, permitiendo que los nodos que no están involucrados en la comunicación vuelvan a dormir,

minimizando la sobre-escucha innecesaria. Como limitante en este mecanismo está su dificultad para la difusión, debido a la inserción de la dirección del receptor en las tramas cortas.

3. Iniciado en el Receptor:

Otro método asíncrono es la transmisión iniciada por el receptor. A diferencia de la técnica de muestreo de preámbulo, en lugar de usar la señalización para dar a conocer que tiene datos para transmitir, el transmisor espera una señalización periódica del receptor y transmite los datos sólo después de que escuche dicha señalización. En este mecanismo, cada nodo se despierta periódicamente, envía una señalización y regresa al estado de reposo, a menos que reciba un reconocimiento, que se refiere a una señal del estado activo del receptor, en cuyo caso comienza a transmitir los datos.

Este mecanismo sustituye el largo preámbulo por las señalizaciones periódicas, lo cual aporta la ventaja de que las señalizaciones del receptor no ocupan el medio durante el largo tiempo del preámbulo del transmisor. Un protocolo que utiliza este mecanismo de ciclo de trabajo de radio es RI-MAC [24].

4. Activación de la radio por demanda:

La activación de la radio por demanda se basa en la idea de que un nodo puede ser retirado del estado de sueño cuando sea necesario. El mecanismo por lo general utiliza otra interfaz de comunicación, llamada: radio de activación, que se refiere a un radio de baja potencia que al escuchar una señal de activación, envía una interrupción a la CPU que activa el radio primario (o radio de datos) como respuesta.

5. Ciclo de trabajo aleatorio:

El ciclo de trabajo aleatorio sigue la idea de que en los despliegues suficientemente densos, los nodos pueden ir a dormir y despertar aleatoriamente, ya que existe una alta probabilidad de que habrá suficientes nodos activos en cualquier momento.

Este mecanismo tiene la limitante de que la aleatoriedad solo se puede utilizar satisfactoriamente en escenarios muy densos, y el ciclo de trabajo debe ser ajustado cuidadosamente a la cantidad de nodos disponibles. Eso significa que los mecanismos de ciclo de trabajo aleatorio necesitan adaptarse a la topología, lo que sacrifica la sencillez que

caracteriza a los mecanismos aleatorios. Las ventajas de este ciclo de trabajo aleatorio incluyen una distribución justa de la carga de tráfico (debido a la aleatoriedad) y un bajo retardo de extremo a extremo (debido a la eliminación de la espera de sueño).

6. Basado en Horarios:

Algunas de las propuestas iniciales de ciclo de trabajo asíncrono se basan únicamente en el diseño de horarios de actividad y de sueño para los nodos sensores. En este mecanismo, los nodos dividen el tiempo en ciclos y cada ciclo en ranuras de tiempo activas e inactivas. No es difícil de ver, por ejemplo, que si dos nodos están activos durante más de la mitad del tiempo, al menos una parte de su tiempo activo se superpone [25], pero tal situación resulta en un ciclo de trabajo demasiado alto, a costa del endeudamiento de otras áreas como el diseño quórum [26] y el diseño de bloques [27].

Para que este mecanismo sea utilizado en el ciclo de trabajo asíncrono, un horario (secuencia de ranuras de tiempo activas e inactivas) debe garantizar que dos nodos superpongan el tiempo activo, independientemente de su desplazamiento (diferencia en su conteo de ranura). Esto se conoce como la propiedad de cierre de rotación [26].

En [28] se demuestra que el horario óptimo, que cumple con la propiedad de cierre de rotación, se basa en diseños de bloques simétricos. El ejemplo de diseño de bloques que presentamos a continuación, se refiere a un ciclo de trabajo simple que consta de 7 ranuras (numeradas de 0 a 6), las ranuras activas son 0,1 y 3 (o cualquier rotación de este horario - $\{0,1,3\}$, $\{1,2,4\}$, $\{2,3,5\}$, $\{3,4,6\}$, $\{4,5,0\}$, $\{5,6,1\}$, $\{6,0,2\}$), los nodos que operan bajo este horario tienen exactamente una ranura activa superpuesta. En [27] se demuestra que el tiempo de superposición es equivalente a la duración de una ranura, independientemente de la alineación de los bordes de la misma. El ciclo de trabajo en este ejemplo es bastante alto (~43%) pero existen otros diseños de bloques que pueden alcanzar ciclos de trabajo de aproximadamente 1%, por ejemplo bloques que contengan miles de ranuras, y de estas pocas estén activas, como el caso, $\{9507, 98, 1\}$.

Dentro de las ventajas de este mecanismo están: la no utilización de mensajes de control, la simple implementación y la independencia de la topología. Su limitación está en que la energía ahorrada se negocia por cantidades significativas de latencia.

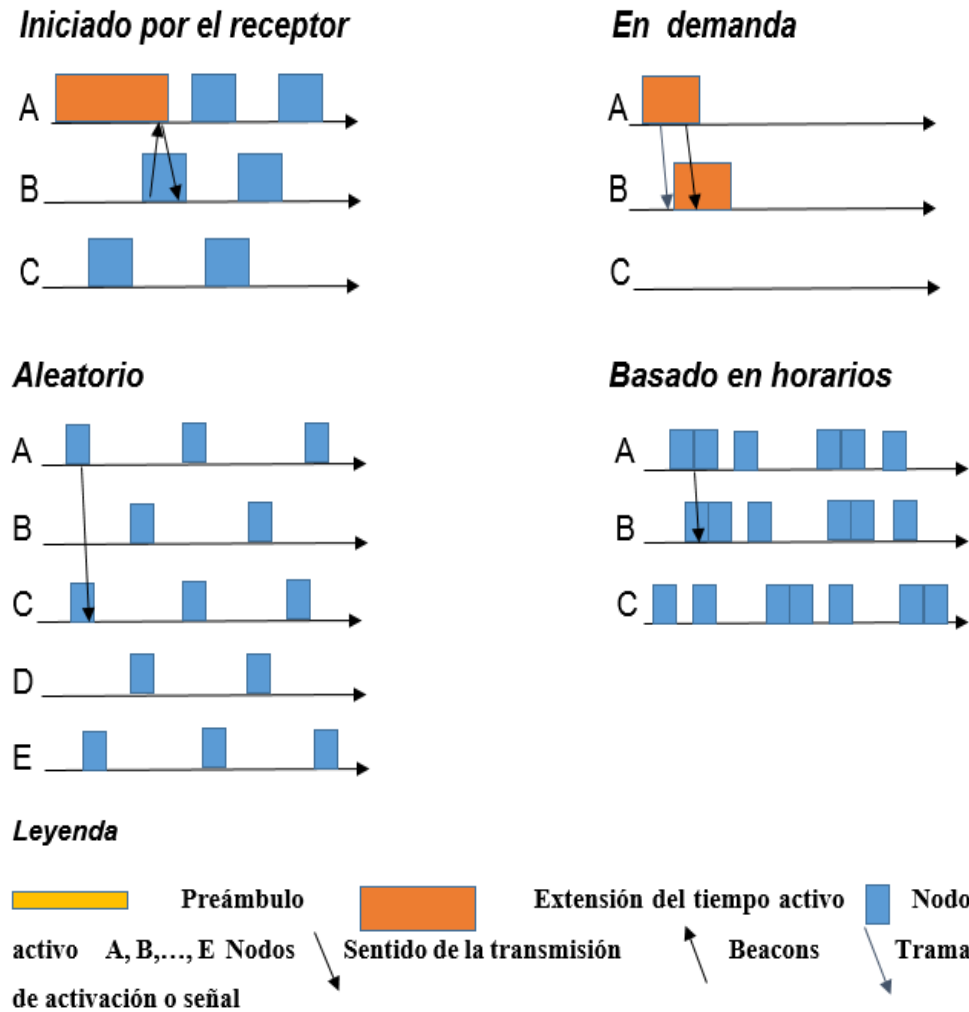


Figura 1.6: Ciclos de Trabajo de Radio Asincrónicos.

1.4 Conclusiones del capítulo

El objetivo fundamental del ciclo de trabajo de radio es reducir el consumo de energía de los nodos sensores y aumentar como consecuencia, el tiempo de vida global de la red.

Las principales desventajas de los ciclos de trabajo se corresponden con la demora del mensaje punto a punto, las tasas de colisión y el sobre-encabezado de control de paquete.

En los ciclos de trabajo de radio sincrónicos los nodos conservan referencias comunes de tiempo. Los mecanismos sincrónicos están dentro de dos esquemas o modelos: el Rendezvous y el Escalonado/Sesgado.

En las propuestas semisíncronas, los nodos son agrupados en clústeres de forma sincrónica y los mismos interactúan entre sí asincrónicamente. En los mecanismos semisíncronos se encuentran el agrupamiento espontáneo dentro del clúster, y el agrupamiento basado en la elección de un nodo coordinador dentro del clúster.

Entre los mecanismos asíncronos encontramos el muestreo de preámbulo, la comunicación iniciada en el receptor, la activación de la radio por demanda, el ciclo de trabajo aleatorio y el basado en horarios.

CAPÍTULO 2. Implementación de los mecanismos de ciclos de trabajo de radio.

En este capítulo se describe el diseño utilizado para la implementación de los mecanismos de ciclo de trabajo de radio: muestreo de preámbulo largo, muestreo de preámbulo corto e iniciado en el receptor, a partir de los protocolos MAC que utilizan dichos mecanismos: B-MAC, X-MAC y RI-MAC, respectivamente.

El capítulo está compuesto por tres epígrafes. En el primer epígrafe se abordan las ventajas de los mecanismos de radio asíncronos que son objeto de estudio en este capítulo. En el segundo epígrafe se hace referencia a los protocolos que se implementan con ayuda del simulador de eventos discretos MATLAB: B-MAC, X-MAC, RI-MAC. En el tercer epígrafe se exponen las principales características de la herramienta de software MATLAB, utilizada para la simulación de los mecanismos de ciclo de trabajo de radio. Además, se muestra el diseño de los mecanismos de ciclo de trabajo de radio mencionados anteriormente, así como los diagramas de flujo de información de las funciones implementadas para cada uno de ellos.

2.1 Ventajas de los ciclos de trabajo de radio asíncronos.

Los ciclos de trabajo asincrónicos ofrecen grandes ventajas respecto a los que requieren de cierta sincronización. Su principal ventaja es que al no necesitar de sincronización, no utilizan mensajes de control y disminuyen la sobre-escucha en gran medida, evidenciando en ahorro energético considerable. Los mecanismos asíncronos son más adaptables a diferentes tipos de topologías. Presentan una mayor escalabilidad pues no necesitan saber si los nodos vecinos entran o salen de la red. Además como no requieren de hardware adicional, disminuyen el costo de los motes.

Como los ciclos de trabajo de radio asíncronos ofrecen mayores ventajas que los síncronos, en este documento son objeto de estudio los protocolos que implementan estos ciclos de trabajo. Se enfatiza en los mecanismos de muestreo de preámbulo largo, muestreo de preámbulo corto e iniciado en el receptor, pues estos son los más utilizados en las RSI, debido a que los demás mecanismos ofrecen características singulares de consumo energético, topología y latencia como el caso de los mecanismos de activación

de la radio por demanda, el ciclo de trabajo aleatorio, y el basado en horarios, respectivamente.

2.2 Protocolos MAC que implementan los Ciclos de Trabajo de Radio Asíncronos.

Entre los protocolos MAC que implementan los mecanismos de ciclo de radio asíncronos encontramos a B-MAC, X-MAC y RI-MAC, que utilizan los mecanismos de muestreo de preámbulo largo, muestreo de preámbulo corto e iniciado en el receptor, respectivamente, que como mencionamos anteriormente son objeto de estudio en este documento.

A continuación se describe el funcionamiento de estos protocolos.

2.2.1 Muestreo de preámbulo largo: B-MAC

En el trabajo de Joseph Pollastre [19] se hace referencia al protocolo para la capa MAC en redes de sensores, nombrado B-MAC, capaz de reducir la denominada escucha ociosa. B-MAC propone que cada nodo despierte periódicamente para comprobar la actividad en el canal, en caso de detección de actividad permanece a la escucha y en caso negativo vuelve a dormir. El tiempo entre periodos activo se fija mediante el denominado intervalo de chequeo. B-MAC define 8 intervalos de chequeo, y cada uno de ellos se corresponde a un modo de escucha diferente.

Para asegurar que todos los paquetes son recibidos, los paquetes se envían con un preámbulo cuya longitud de transmisión es superior al intervalo de chequeo. B-MAC define 8 tamaños de preámbulo, cada uno relacionado con un diferente modo de transmisión.

Otra gran ventaja de B-MAC es su modularidad y flexibilidad, B-MAC proporciona interfaces accesibles a las capas superiores, con la finalidad que desde estas capas se puedan fijar los modos de escucha y transmisión, ajustándose a las necesidades del momento. Su propio autor refleja que usando dichas interfaces, y realizando las modificaciones de modo oportunas, se pueden llegar a conseguir grandes ahorros de energía en la red.

2.2.2 Muestreo de preámbulo corto: X-MAC

X-MAC [23] utiliza un preámbulo, en el cual se transmite una serie de tramas cortas intermitentes, en ellas se encuentra la dirección del nodo receptor, dificultando esto la radiodifusión en la red. Los pequeños intervalos entre las tramas, permiten al nodo receptor enviar un reconocimiento al transmisor, indicándole que interrumpa el preámbulo y comience a transmitir los datos.

Con la interrupción del preámbulo se ahorra energía tanto en el transmisor como en el receptor y logra una baja latencia. El nodo receptor no involucrado en la comunicación, vuelve al modo de sueño inmediatamente, en lugar de quedarse escuchando el preámbulo completo, disminuyendo así la sobre-escucha.

2.2.3 Iniciado por el receptor: RI-MAC

Mientras LPP (del inglés, *Low Power Processor*) se concibió como un medio para despertar a una red de tráfico configurable, RI-MAC [24] utiliza la comunicación iniciada en el receptor como una solución más genérica, que opera en un amplio rango de patrones y de tasas de tráfico.

El protocolo RI-MAC tiene el siguiente funcionamiento: un nodo se despierta periódicamente para chequear si tiene mensajes de señalización (beacons), si tiene mensajes pendientes para transmitir debe esperar despierto y comenzar su transmisión inmediatamente después de recibir el beacon del receptor, el receptor envía un ACK como reconocimiento del paquete recibido y lo invita a una nueva transmisión de datos; el transmisor regresa al estado de sueño después de transmitir el dato y verificar que no tiene más datos que transmitir. Este protocolo reduce en gran medida la ocupancia del medio por los nodos antes que encuentren un periodo de actividad común para transmitir los datos.

Al igual que en el muestreo de preámbulo, la carga del gasto extra de energía se encuentra en el transmisor, pues este permanece activo hasta la recepción de la señalización del receptor previsto, pero RI-MAC aporta mejoras significativas respecto a X-MAC, en particular cuando el tráfico es mayor.

2.3 Diseño utilizando la herramienta de software MATLAB de los mecanismos de ciclo de trabajo de radio

Como se introdujo en el Capítulo I, en este epígrafe se analizan los mecanismos de tipo asincrónico por la ventaja que estos experimentan sobre los de tipo sincrónico. Los mecanismos seleccionados para la simulación fueron los de muestreo de preámbulo largo y corto y el mecanismo iniciado en el receptor, que son de gran utilidad para las redes de sensores inalámbricos en la actualidad.

2.3.1 Herramienta de Software MATLAB

Para la implementación de los mecanismos de ciclo de trabajo se utiliza el software MATLAB, debido a que no se recurre a todas las herramientas de los simuladores de propósitos específicos, sino que se cuenta con una red simple, de un solo salto para la comunicación entre los nodos transmisor y receptor, además se analizan las características que presentan los protocolos respecto al ciclo de trabajo de radio y no otros parámetros de la red como el enrutamiento, el rendimiento de las redes multisalto, entre otros.

MATLAB es un programa de cálculo técnico y científico. Es el nombre abreviado de “MATrix LABoratory”. MATLAB es un programa para realizar cálculos numéricos con vectores y matrices. Una de las capacidades más atractivas es la de realizar una amplia variedad de gráficos en dos y tres dimensiones. MATLAB tiene también un lenguaje de programación propio. Este lenguaje de programación es una herramienta de alto nivel, necesaria para desarrollar aplicaciones técnicas, es fácil de utilizar y asegura significativamente el aumento de la productividad de los programadores, respecto a otros entornos de desarrollo. En nuestro caso lo utilizamos en la simulación de ciclos de trabajo de radio a partir de ficheros de texto ASCII, con la extensión *.m, que contienen conjuntos de comandos o definición de funciones [29].

2.3.2 Diseño del Mecanismo de Ciclo de Trabajo de Radio: Muestro de preámbulo largo.

Para la realización del diseño del Mecanismo de Ciclo de Trabajo de Radio: Muestro de preámbulo largo, se toma como referencia el protocolo MAC para las RSI: B-MAC, que utiliza el mecanismo ya mencionado.

B-MAC [20] fue creado para aplicaciones de monitoreo medioambiental. La principal ventaja de este mecanismo es que disminuye significativamente el consumo energético respecto a los mecanismos síncronos, evidenciando la mejora de utilizar los protocolos asíncronos en aplicaciones que no requieran de sincronismo.

El conjunto de instrucciones para este programa (Anexo I), se inicia a partir de una función llamada `preamble_sampling`, para identificar el mecanismo de muestro de preámbulo largo. En la Figura 2.1 se muestra el diagrama de flujo de información del mecanismo implementado.

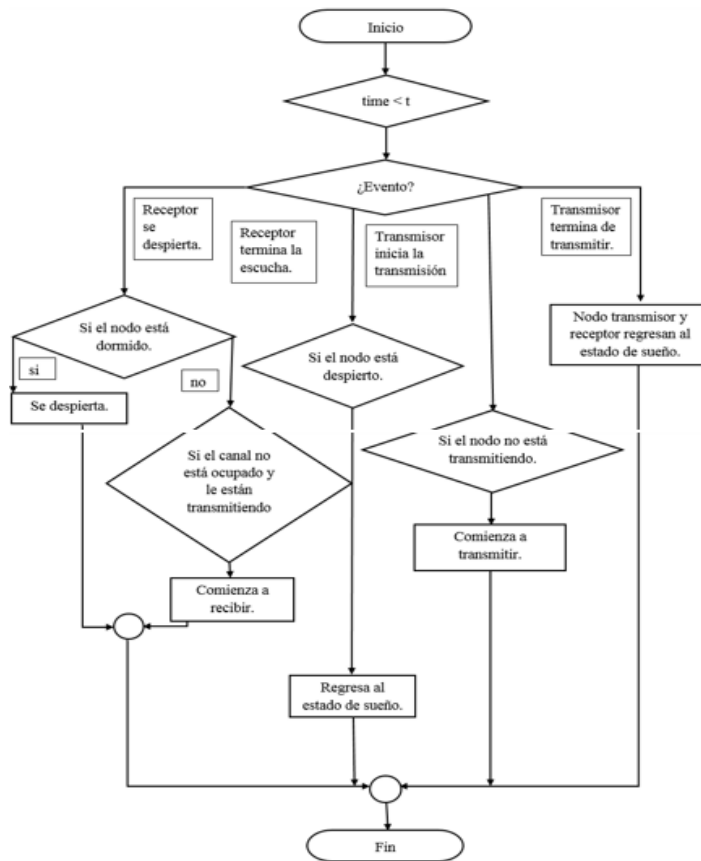


Figura 2.1: Diagrama de flujo de información de la función `preamble_sampling`.

2.3.3 Diseño del Mecanismo de Ciclo de Trabajo de Radio: Muestro de preámbulo corto.

Para la implementación del Mecanismo de Ciclo de Trabajo de Radio: Muestreo de preámbulo corto, se toma como referencia el protocolo X-MAC, el cual es un protocolo de la capa MAC que utiliza este tipo de mecanismo [23].

La función implementada que describe el comportamiento de este mecanismo de ciclo de trabajo de radio, se denomina `int_preamble_sampling` (Anexo II).

Con la interrupción del preámbulo corto por el receptor, este mecanismo logra aumentar el tráfico de la red, respecto al mecanismo de preámbulo largo. En la figura 2.2 se presenta el diagrama de flujo de información de la función `int_preamble_sampling`, mencionada anteriormente.

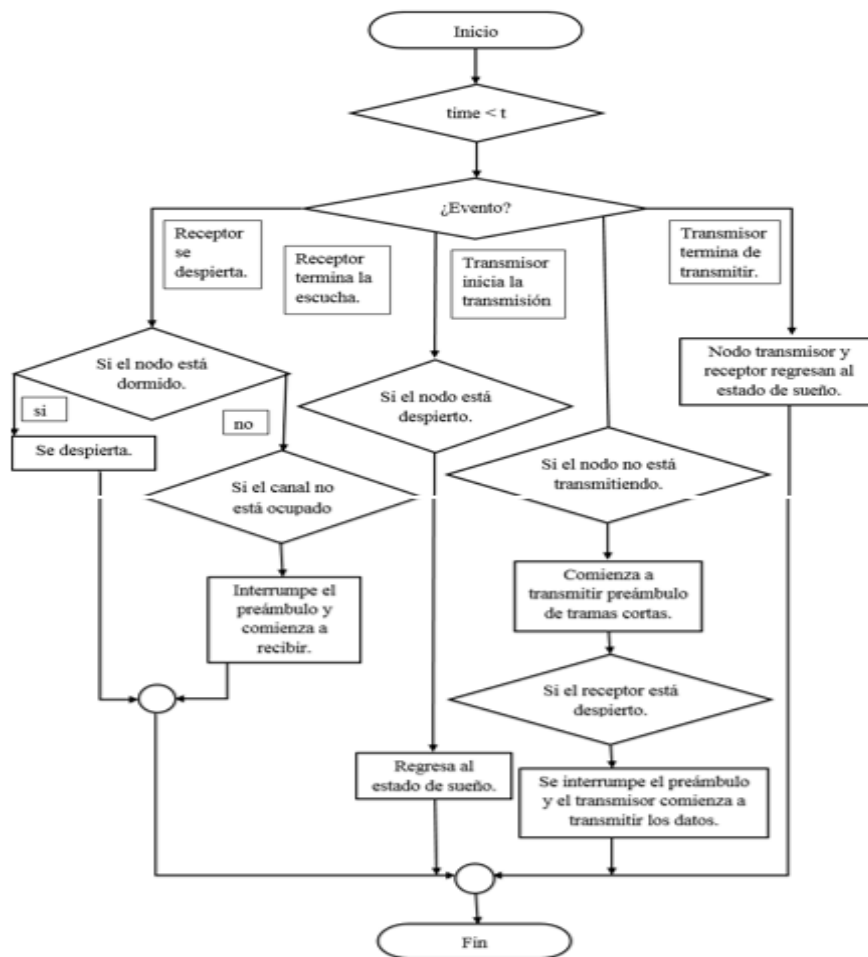


Figura 2.2: Diagrama de flujo de información de la función `int_preamble_sampling`.

2.3.4 Diseño del Mecanismo de Ciclo de Trabajo de Radio: Iniciado en el receptor.

Para la puesta en práctica de la simulación de este protocolo se utiliza el principio de funcionamiento del protocolo de la capa MAC: RI-MAC [24], cuyo mecanismo de ciclo de trabajo de radio es el analizado a continuación.

El código de programa que describe el funcionamiento de este mecanismo, se implementa a partir de la función `inic_rx` (Anexo III).

Este mecanismo se elaboró pensando en un mejor comportamiento de la latencia para tráfico pesado en la red, respecto a los mecanismos precedentes. En la figura 2.3 se observa el diagrama de flujo de información de este mecanismo a partir de la función implementada.

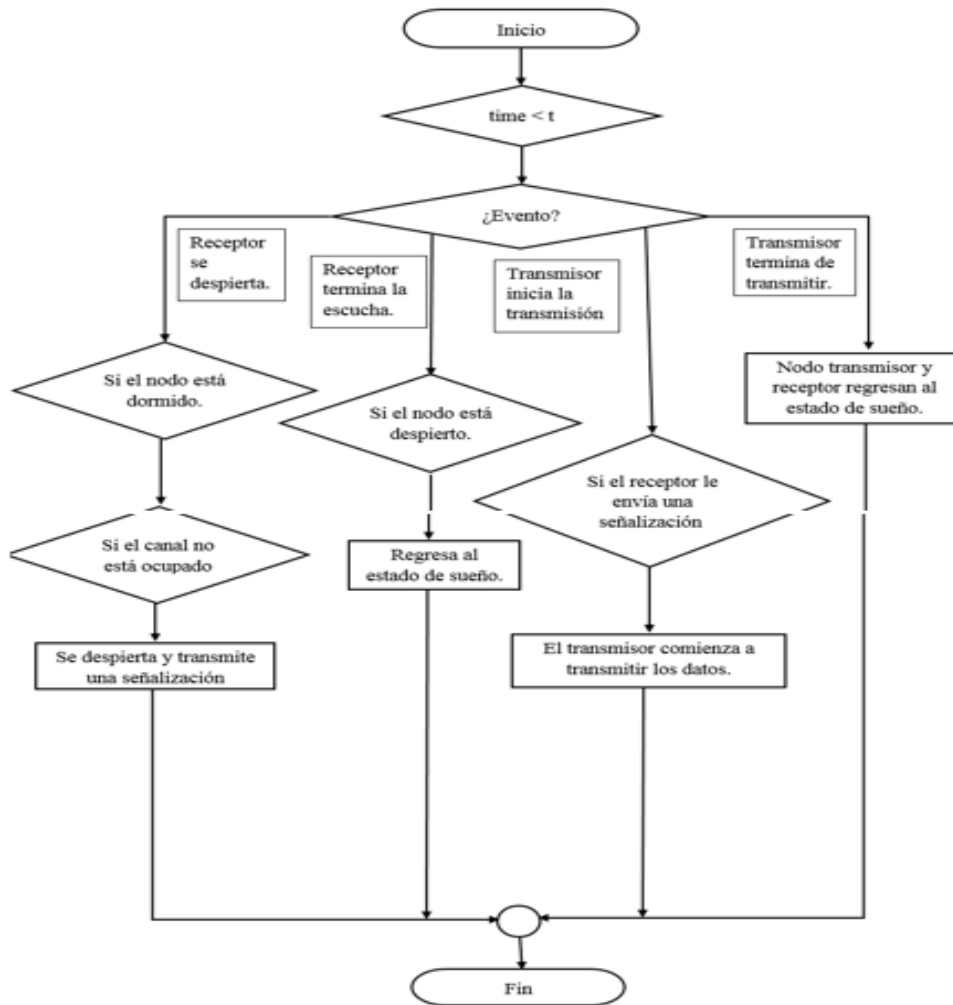


Figura 2.3: Diagrama de flujo de información de la función `inic_rx`.

2.4 Conclusiones del capítulo.

Entre las ventajas de los mecanismos asíncronos sobre los que requieren de sincronismo están: el menor consumo energético, la mayor adaptabilidad a la topología de la red, el no requerimiento de hardware adicional, la no utilización del sobre-encabezado de control y la mayor escalabilidad de la red.

Los protocolos de la capa MAC: B-MAC, X-MAC y RI-MAC utilizan los mecanismos asíncronos de muestreo de preámbulo largo, muestreo de preámbulo corto e iniciado en el receptor respectivamente.

Los mecanismos asíncronos de muestreo de preámbulo largo, muestreo de preámbulo corto e iniciado en el receptor son los más utilizados en las RSI debido a su mejor funcionamiento frente a las aplicaciones actuales, pues los demás mecanismos responden a características específicas de consumo energético, densidad de la red y latencia en la transmisión de datos.

Para el diseño de los mecanismos de ciclo de trabajo de radio se toman como referencia, los protocolos MAC que aplican los ciclos de trabajo analizados: B-MAC, X-MAC, y RI-MAC.

CAPÍTULO 3. Resultados y discusión acerca de los mecanismos de ciclo de trabajo de radio simulados.

Los mecanismos de ciclo de trabajo simulados: muestreo de preámbulo largo, muestreo de preámbulo corto e iniciado en el receptor, presentan diferentes comportamientos en cuanto a las siguientes medidas de desempeño: consumo energético, latencia y tasa de transmisión. En este capítulo analizaremos estos comportamientos para cada uno de los mecanismos, estableciendo comparaciones entre ellos. En el proceso de análisis de las medidas de desempeño, se realizan experimentos para valorar el comportamiento de las estrategias ante los cambios de la red. De esta forma se evalúa la escalabilidad, la capacidad de soportar diferentes cargas de tráfico y la incidencia del ciclo de trabajo, en el funcionamiento de los ciclos de trabajo de radio seleccionados.

En el primer epígrafe de este capítulo se describe el escenario de simulación propuesto para cada mecanismo de ciclo de trabajo de radio. En el segundo epígrafe se observa el comportamiento de las medidas de desempeño para los tres protocolos propuestos variando el número de nodos. En el tercer epígrafe se muestra el comportamiento de las medidas de desempeño variando el tráfico de la red y en el cuarto epígrafe se expone el comportamiento de las medidas de desempeño variando el ciclo de trabajo.

3.1 Descripción del escenario de simulación

El escenario de simulación inicial propuesto para la red de sensores inalámbricos, tiene las siguientes características: la topología representada es poco densa, pues cuenta con la participación de 25 nodos, el tráfico es ligero con una media del espaciamiento en la transmisión de paquetes de 60 unidades de tiempo, los paquetes que se envían son de tamaño fijo y equivalente a 5 unidades de tiempo y el ciclo de trabajo establecido es del 1%.

Para estimar las medidas de desempeño se realizan 5 simulaciones con una duración 3000 unidades de tiempo cada una. Estos valores están supeditados a limitaciones en el hardware utilizado para correr las simulaciones y el tiempo que se requiere para realizar las mismas. Los valores que se presentan se obtuvieron promediando los resultados de las 5

simulaciones independientes. Las tres medidas evaluadas (consumo, latencia y tasa de transmisión) se expresan en sus unidades correspondientes por nodo.

3.2 Experimento variando el número de nodos

En este experimento se relacionan las medidas de desempeño de los ciclos de trabajo de radio con la variación del número de nodos dentro de la red.

La primera medida de desempeño a analizar es el consumo energético de los nodos sensores, para el cálculo de la energía consumida se toma en consideración una resistencia de carga normalizada a 1Ω , y los nodos sensores de tipo Imote2, los cuales presentan las características energéticas de la Tabla 3.1:

Tabla 3.1. Características del hardware del nodo sensor Imote2 (tomado del data sheets por el fabricante).

Mota/Modelo de radio	Radio			Modelo	CPU	
	Flujo de Corriente				Flujo de corriente	
	Sueño	Escucha	Activo Tx/Rx		Sueño	Activo
Imote2/ TICC2420	20 μ A	0.426mA	15mA	Intel PXA271	390 μ A	66mA

En la Figura 3.1 se observa el comportamiento del consumo energético a medida que aumenta el número de nodos.

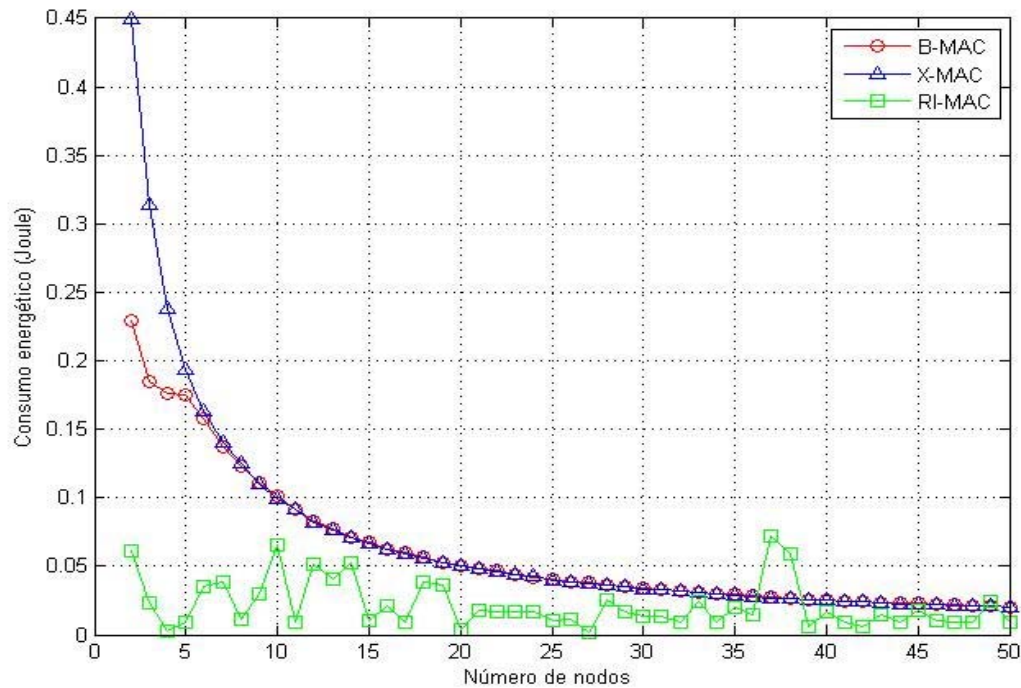


Figura 3.1: Consumo energético a medida que aumenta el número de nodos.

Como se puede observar en la Figura 3.1, para los protocolos B-MAC y X-MAC el consumo energético por nodo disminuye, a medida que aumentamos la densidad de la misma. En el caso de una red compuesta de pocos nodos se observa que el protocolo X-MAC tiene mucho mayor consumo que el protocolo B-MAC debido a que en este logra transmitir una mayor cantidad de paquetes, consumiendo así mayor energía. El protocolo RI-MAC muestra variaciones en el consumo energético en función del número de nodos en la red, manteniendo un bajo consumo respecto a los demás protocolos representados. Esto se debe fundamentalmente a que minimiza el tiempo que tardan los nodos transmisor y receptor para hacer coincidir sus periodos de modo activo para la comunicación, además de no utilizar ningún tipo de preámbulo durante la transmisión.

Como segunda medida de desempeño analizamos la latencia o demora en la transmisión de paquetes. En la Figura 3.2 se muestra la latencia por paquete transmitido a medida que aumenta el número de nodos.

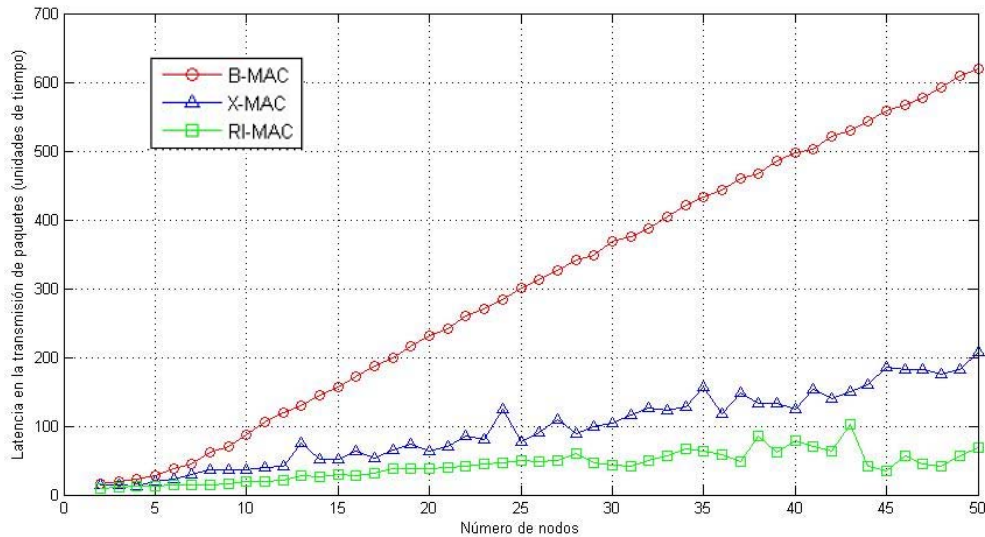


Figura 3.2: Demora por paquete a medida que aumenta el número de nodos.

En la figura anterior se observa como la demora en la transmisión exitosa del paquete en los protocolos B-MAC y X-MAC aumentan a medida que aumentan los nodos sensores dentro de la red, mientras que el protocolo RI-MAC aumenta ligeramente su latencia. El mejor comportamiento con el aumento de la densidad de la red lo tiene el protocolo RI-MAC, esto se debe al igual que en el caso anterior, al menor tiempo utilizado para la conexión de los nodos transmisor y receptor. En los casos de los protocolos B-MAC y X-MAC presentan mayor latencia por el uso de transmisión de preámbulo. Siendo B-MAC el de peor comportamiento debido a su largo preámbulo con respecto a X-MAC. El aumento de la latencia con la densidad de la red se debe en todos los casos a la mayor ocupación del canal, lo que trae consigo que los nodos tengan que esperar mayor tiempo para la transmisión

Por último se establece el comportamiento de la tasa de transmisión. En la Figura 3.3 se observa el comportamiento de la tasa de transmisión en los protocolos B-MAC, X-MAC y RI-MAC a medida que aumenta el número de nodos.

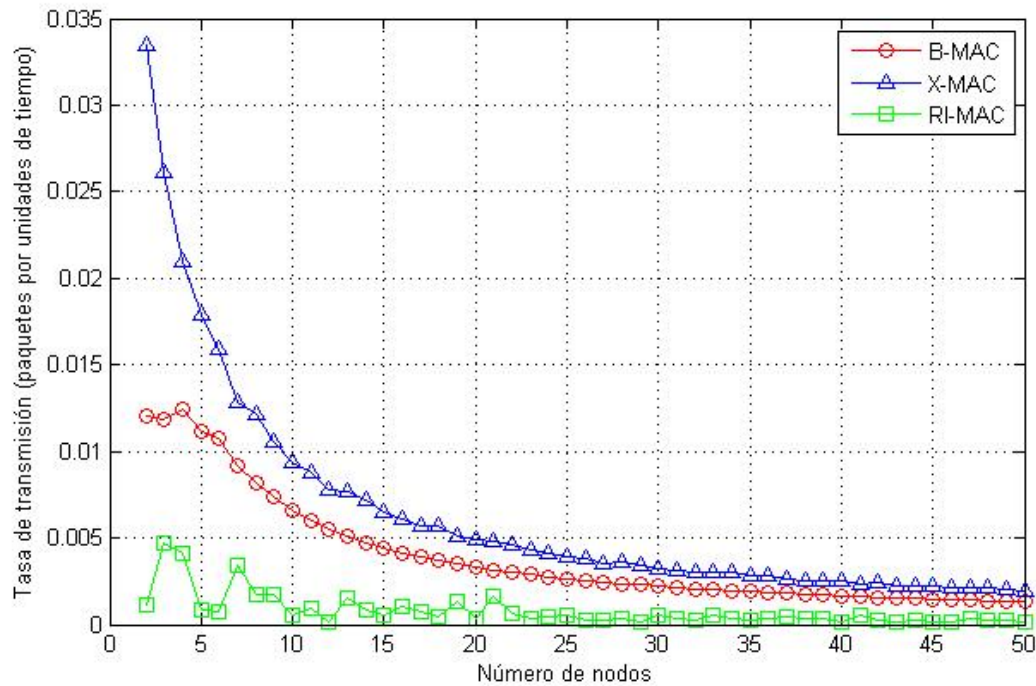


Figura 3.3: Tasa de transmisión de la red a medida que aumenta el número de nodos.

Podemos observar en la Figura 3.3 como a medida que aumentamos la densidad de la red, la tasa de transmisión en cada nodo disminuye, debido a que el canal tiene capacidad fija que debe ser compartida entre todos los nodos. La mayor tasa de transmisión la tiene el protocolo X-MAC, pues este protocolo fue optimizado para cargas de tráfico ligeras, como el caso representado. El protocolo B-MAC presenta menor tasa de transmisión respecto a X-MAC debido al largo preámbulo en la transmisión de los datos, aumentando la latencia por envío de paquetes y por tanto la cantidad de paquetes transmitidos disminuye. En cuanto al protocolo RI-MAC, vemos un comportamiento singular para esta red simulada, pues es un protocolo con una gran eficiencia en cuanto a la latencia y el consumo, sin embargo vemos como en este caso presenta una baja tasa de transmisión. Esto se debe a que en el mismo tiempo de simulación este protocolo mantiene un balance más o menos constante de tasa de transmisión por nodo, independiente de la cantidad de nodos de la red. En una red real este balance de la transmisión posibilita una baja probabilidad de perder los paquetes, incluso logra la transmisión desde nodos ocultos exitosamente con una baja latencia.

3.3 Experimento variando el tráfico

En este experimento se evidencia el rendimiento de los protocolos en cuanto a las medidas de desempeño con la disminución de la media del espaciamiento entre los paquetes que se desean transmitir, o sea, que se aumenta el tráfico en la red.

La medida de desempeño que se analiza a continuación es el rendimiento energético. En la Figura 3.4 se muestra el consumo energético de cada nodo a medida que aumenta el tráfico.

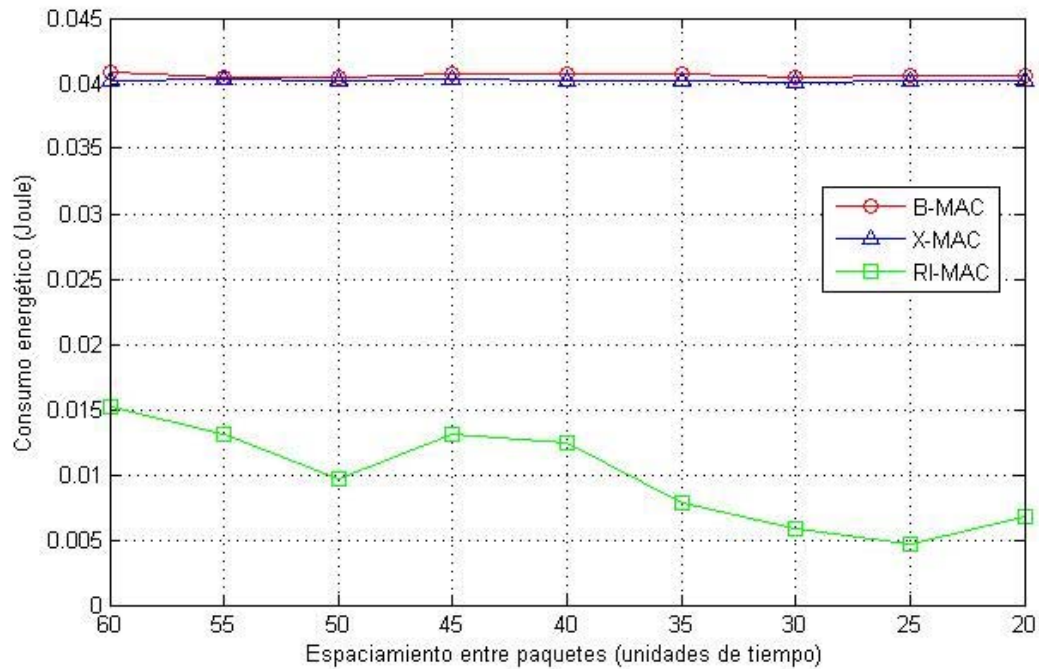


Figura 3.4: Consumo energético a medida que aumenta el tráfico.

En el caso del consumo energético podemos observar como los protocolos B-MAC y X-MAC se mantienen prácticamente constantes a medida que varía el tráfico, siendo el protocolo B-MAC quien presenta el mayor consumo debido a la transmisión de su largo preámbulo. Mientras, el consumo de RI-MAC se manifiesta de manera descendente, evidenciando su principal característica que se refiere al alto rendimiento energético para una carga de tráfico dinámica, sobre todo para cargas de tráfico críticas como los flujos de datos en ráfaga. A medida que aumenta el tráfico el protocolo RI-MAC aumenta su rendimiento energético.

La siguiente medida de desempeño que se analiza es la latencia en la transmisión de paquetes. En la Figura 3.5 se muestra el comportamiento de la demora en la transmisión de paquetes, con el aumento del tráfico.

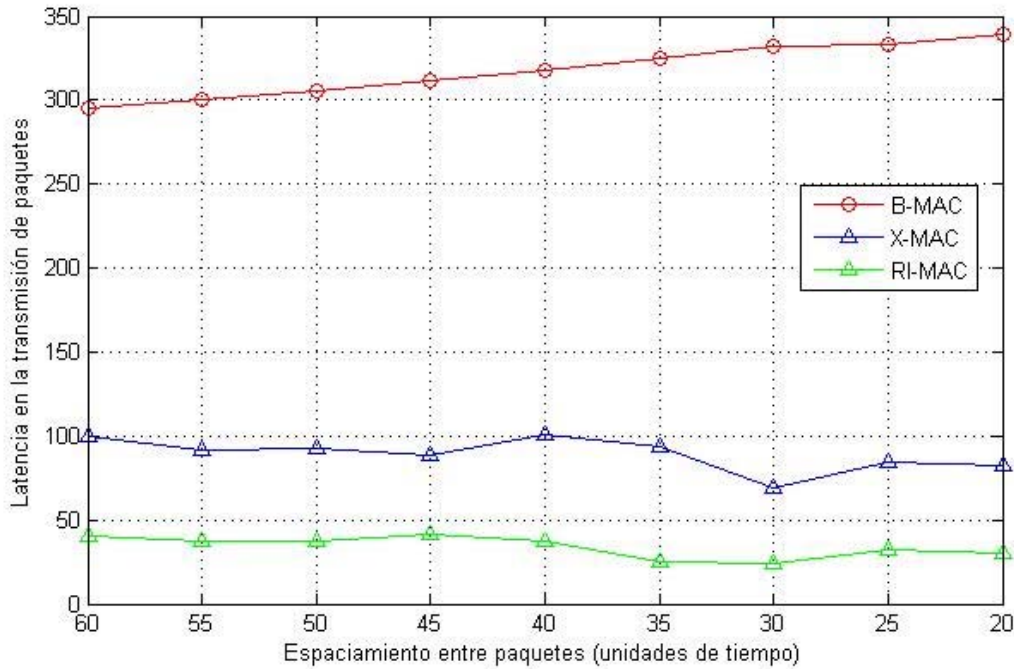


Figura 3.5: Demora en la transmisión del paquete a medida que aumenta el tráfico.

Como se observa en la figura anterior, a medida que aumentamos el tráfico, vemos como el protocolo B-MAC presenta el peor comportamiento, pues con la transmisión del largo preámbulo en conjunto con los datos, aumenta su latencia. El protocolo X-MAC presenta menor latencia que B-MAC pues este transmite una secuencia de preámbulos cortos que pueden ser interrumpidos por el receptor disminuyendo su latencia. El protocolo RI-MAC mantiene un bajo consumo, debido al menor tiempo utilizado por este para la conexión de los nodos transmisor y receptor, evidenciando su alto rendimiento en cuanto a la latencia en un amplio rango de carga de tráfico.

Por último se observa en la Figura 3.6, el comportamiento de los tres protocolos en cuanto a la tasa de transmisión a medida que aumenta el tráfico en la red.

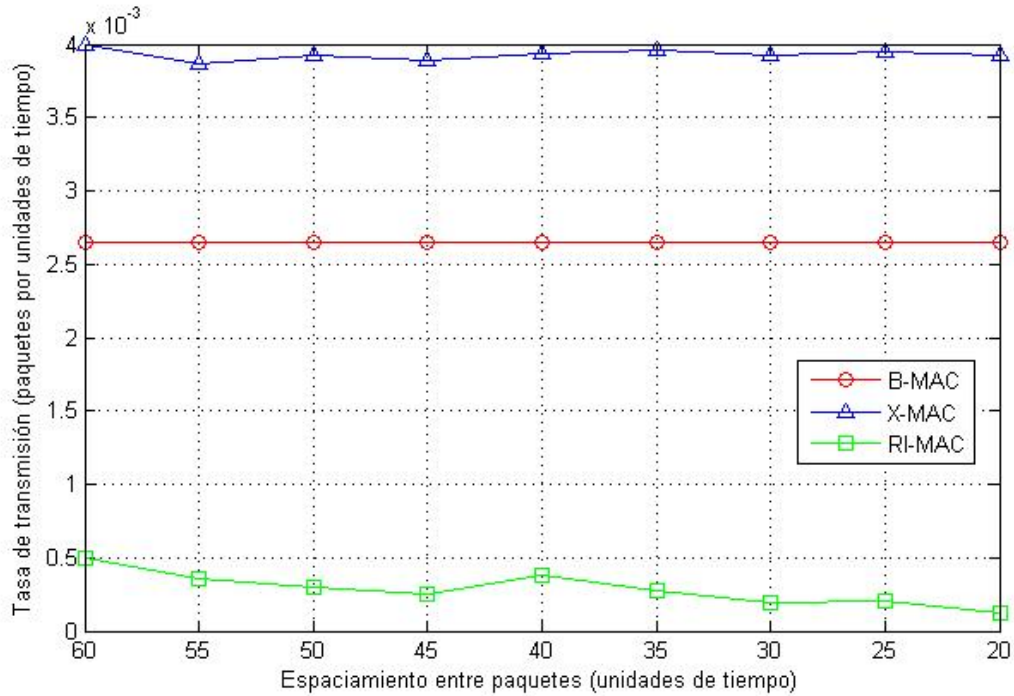


Figura 3.6: Tasa de transmisión de la red a medida que aumenta el tráfico.

En este caso se observa que el protocolo X-MAC presenta mayor tasa de transmisión que B-MAC, debido a que transmite un preámbulo corto, por lo que tiene menor demora en el envío de paquetes, aumentando el número de paquetes transmitidos, mientras que el protocolo B-MAC presenta un preámbulo en la transmisión de los datos mucho mayor, transmitiendo menos paquetes. En el caso de RI-MAC su tasa de transmisión se mantiene baja respecto a los demás protocolos la cual se corresponde con su comportamiento descrito en la sección anterior.

3.4 Experimento variando el ciclo de trabajo

En este experimento se describe el comportamiento de las medidas de desempeño variando el ciclo de trabajo de radio.

Primeramente se analiza el consumo energético. En la Figura 3.7 se muestra el consumo energético a medida que aumentamos el ciclo de trabajo de radio en el rango de los valores típicos en las RSI con ciclos de trabajo asíncronos (1%-5%).

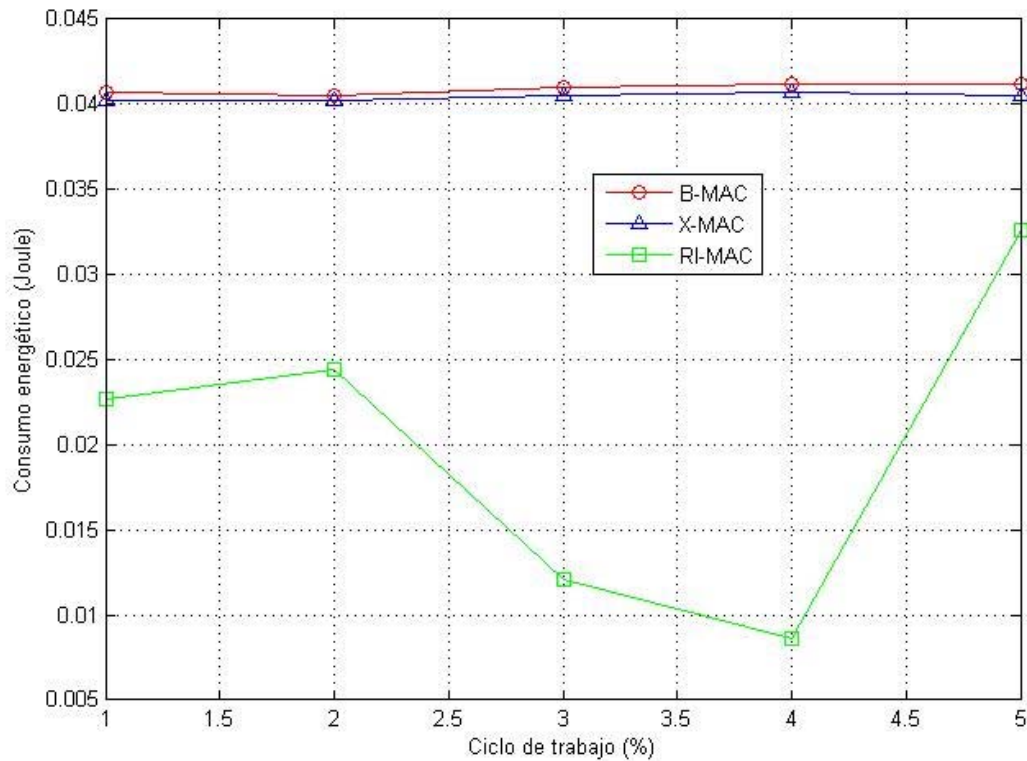


Figura 3.7: Consumo energético a medida que aumenta el ciclo de trabajo.

En la Figura 3.7 vemos que el consumo energético en el protocolo B-MAC es mayor comparado con X-MAC a medida que aumentamos el ciclo de trabajo, debido al largo preámbulo que debe transmitir junto a los datos. En ambos casos el incremento del consumo con el aumento del ciclo de trabajo es prácticamente imperceptible. En cuanto al protocolo RI-MAC varía su consumo energético con la variación del ciclo de trabajo, manteniendo un bajo consumo respecto a los demás protocolos representados, debido al menor tiempo utilizado en el logro de la conexión entre los nodos transmisor y receptor, y la no utilización de preámbulo.

A continuación se observa en la Figura 3.8, el comportamiento de la latencia en la transmisión de paquetes a partir de la variación del ciclo de trabajo.

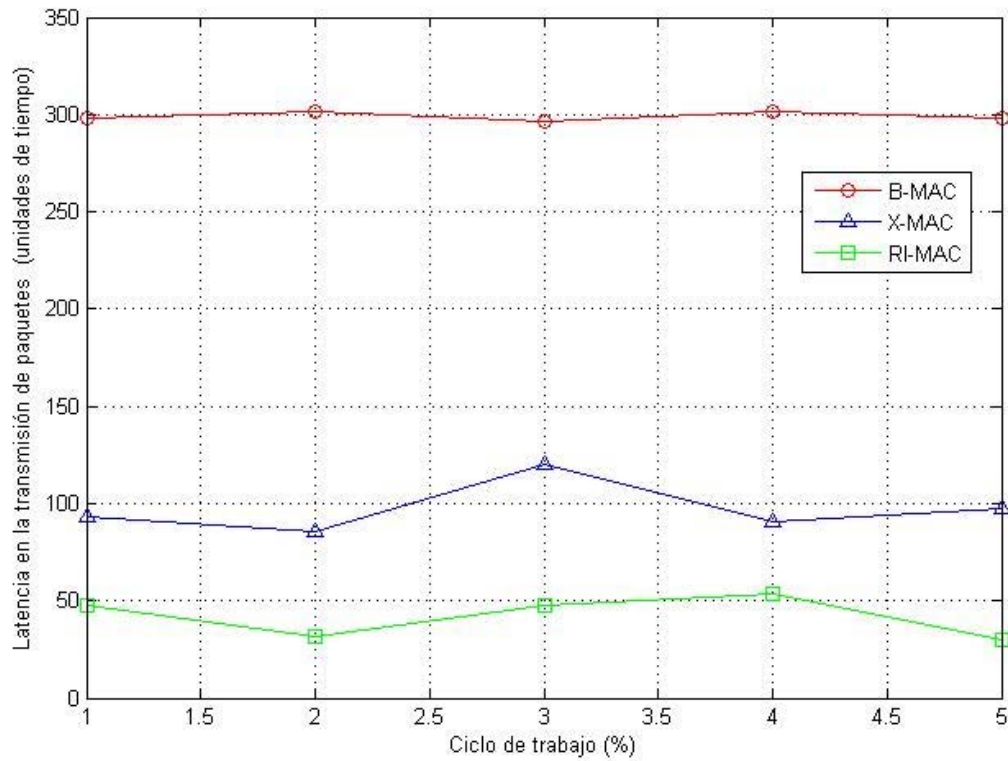


Figura 3.8: Demora en la transmisión del paquete a medida que aumenta el ciclo de trabajo.

Como se puede ver en la Figura 3.8, la mayor latencia la tiene el protocolo B-MAC, mientras las demoras de los protocolos X-MAC y RI-MAC son menores. El protocolo X-MAC presenta menor preámbulo respecto a B-MAC. Mientras, RI-MAC utiliza menor tiempo en el logro de la comunicación.

La última medida de desempeño a analizar variando el ciclo de trabajo de radio, es la tasa de transmisión. En la Figura 3.9 veremos el comportamiento de este parámetro.

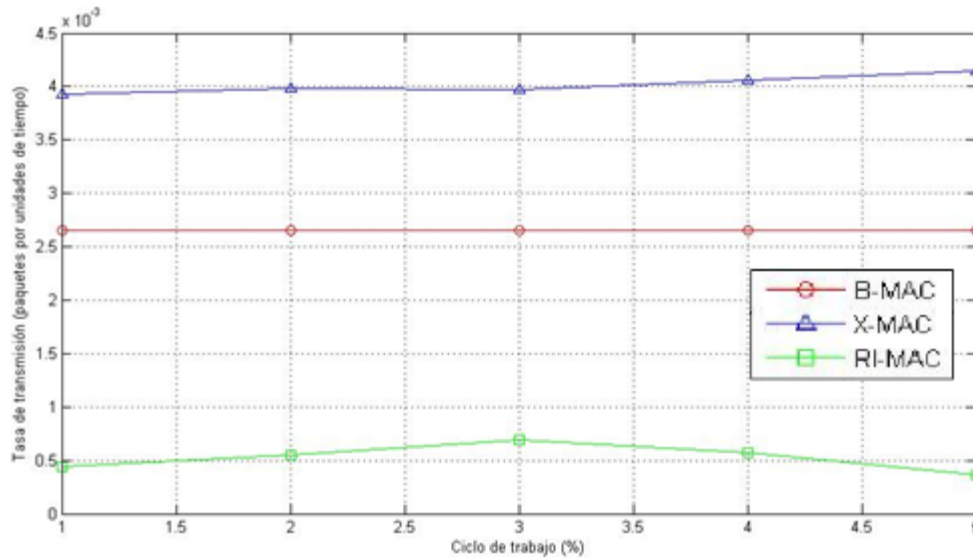


Figura 3.8: Tasa de transmisión de la red a medida que aumenta el ciclo de trabajo.

En este último caso se muestra como el protocolo X-MAC tiene la mayor tasa de transmisión. El protocolo B-MAC muestra una tasa de transmisión constante con el aumento del ciclo de trabajo y menor respecto a X-MAC debido a que transmite un preámbulo mayor que el protocolo X-MAC. Este preámbulo es una secuencia de tramas cortas interrumpibles para comenzar la transmisión. El protocolo RI-MAC con la variación del ciclo de trabajo tiene la menor tasa de transmisión. Otra vez vemos que el protocolo X-MAC presenta los mejores resultados en cuanto a la utilización de la capacidad del canal de comunicaciones.

3.5 Conclusiones del capítulo.

El protocolo B-MAC presenta el mayor consumo energético, y mayor latencia en la transmisión de paquetes, debido al largo preámbulo que transmite antes de los datos para lograr la transmisión exitosa de paquetes hacia el nodo receptor.

El protocolo X-MAC presenta el menor consumo energético, y latencia comparado con B-MAC, pues transmite una secuencia de tramas cortas a modo de preámbulo, la cual puede ser interrumpida por el receptor para recibir los datos, consumiendo así menos energía y menos tiempo. Además X-MAC es el que mejor tasa de transmisión alcanza en todos los casos.

El protocolo RI-MAC mantiene un bajo consumo y latencia en la transmisión de los datos con respecto a los demás protocolos analizados, pues es el receptor el que decide cuando comenzar la transferencia de datos, logrando un menor tiempo en la espera del modo activo, tanto del nodo transmisor como del receptor para comenzar la comunicación.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- En esta investigación se caracterizaron los principales mecanismos propuestos para el empleo de ciclos de trabajo de radio, definiéndose como criterio de clasificación el uso o no de técnicas de sincronismo. Identificándose como medidas de desempeño fundamentales de los protocolos MAC para las RSI, el consumo energético, la latencia en la transmisión de paquetes, y la tasa de transmisión.
- Apañándose en la revisión realizada se puede concluir que los mecanismos de ciclo de trabajo de radio asíncronos logran un menor consumo energético que los síncronos y semisíncronos.
- Se implementaron con ayuda del simulador de eventos discreto, el software MATLAB, los protocolos B-MAC, X-MAC, y RI-MAC. Los cuales emplean los mecanismos de preámbulo largo, muestreo de preámbulo corto, e iniciado en el receptor respectivamente.
- Como resultado de la comparación entre estos protocolos se evidencia lo siguiente:
 - 1 El protocolo B-MAC presenta un alto consumo energético y latencia debido a la transmisión de un largo preámbulo que precede a la transmisión de datos.
 - 2 El protocolo X-MAC presenta la mejor tasa de transmisión en todos los escenarios analizados.
 - 3 En todos los escenarios el protocolo RI-MAC supera a B-MAC y X-MAC en cuanto a consumo energético y latencia. Obteniendo valores de estos indicadores que en la mayoría de los casos están por debajo del 50% de los alcanzados por B-MAC y X-MAC.

Recomendaciones

- 1 Incorporar nuevos mecanismo de ciclo de trabajo de radio en las comparaciones. Así como otros parámetros de desempeño tales como el número de colisiones que se producen en la red.
- 2 Utilizar simuladores de propósito específico y hardware más potente en los estudios, de forma que se puedan evaluar escenarios más complejos durante mayor tiempo.
- 3 Extender el estudio mediante la inclusión de protocolos de otras capas como es el caso de los protocolos de enrutamiento y evaluar su interacción con los mecanismos de ciclo de trabajo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Thumann and D. Mehta, Handbook of energy engineering. Fairmont Press, 2008.
- [2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Netw.*, vol. 7, pp. 537–568, May 2009.
- [3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” *SIGPLAN Not.*, vol. 35, pp. 93–104, November 2000.
- [4] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” in *Proc. 29th Annual IEEE International Conference on Local Computer Networks*, November 2004, pp. 455–462.
- [5] Carrano, R.C.; Passos, D.; Magalhaes, L.C.S.; Albuquerque, C.V.N., "Survey and Taxonomy of Duty Cycling Mechanisms in Wireless Sensor Networks," *Communications Surveys & Tutorials, IEEE* , vol.16, no.1, pp.181,194, First Quarter 2014
- [6] Z. Teng and K.-I. Kim, “A Survey on Real-Time MAC Protocols in Wireless Sensor Networks,” *Communications and Network*, vol. 02, pp. 104–112, 2010.
- [7] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing”. 7 th annual international conference on Mobile computing and networking, July 16-21, 2001, Rome, Italy. pp. 70-84.
- [8] V. Raghunathan, C.Schurgers, S. Park and M.B. Srivastava, "Energy-aware wireless microsensor networks," *Journal of IEEE Signal Processing Magazine*, vol.19, no.2, pp.40-50, Mar 2002.
- [9] R. Amirtharajah, S. Meringer, J. O. Mur-Miranda, A. Chandrakasan and J. Lang, “A Micropower Programmable DSP Powered using a MEMSbased Vibration-to-Electric Energy Converter,” 5 th IEEE Symposium on Computers and Communications (ISCC 2000). Vol. 43, pp. 362-363, February, 2000.

- [10] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks", Special Interest Group on Management Of Data (SIGMOD) 2002, Vol 31, No 3, September 2002.
- [11] Y. Sankarasubramaniam, I. F. Akyildiz and S. W. McLaughlin, "Energy efficiency based packet size optimization in wireless sensor networks", First IEEE Workshop on Sensor Network Protocols and Applications (SNPA 2003). Anchorage, Alaska, USA, May 2003.
- [12] R. Serna Oliver and G. Fohler, "A proposal for a notion of timeliness in wireless sensor networks," in Proceedings of the 8th International Workshop on Real-Time Networks RTN'09, June 2009.
- [13] C. Peraki, S. D. Servetto. "On the Maximum Stable Throughput Problem in Random Networks with Directional Antennas," in Proc. of the 4th ACM MobiHoc, June 2003.
- [14] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," IEEE Network, vol. 18, no. 4, pp. 45–50, July-August 2004.
- [15] G. Lu, B. Krishnamachari, and C. Raghavendra, "An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks," in Proc. 18th International Parallel and Distributed Processing Symposium, April 2004, pp. 224–235.
- [16] S. Parmar, S. Nandi, and A. Chowdhury, "Power efficient and low latency MAC for wireless sensor networks," in Proc. 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06), vol. 3, September. 2006, pp. 940–944.
- [17] S. Hussain, T. Khan, and S. Zaidi, "Latency and energy efficient MAC (LEEMAC) protocol for event critical applications in wsns," in Proc. International Symposium on Collaborative Technologies and Systems (CTS 2006), May 2006, pp. 370–378.
- [18] L. Choi, S. H. Lee, and J.-A. Jun, "SPEED-MAC: Speedy and energy efficient data delivery MAC protocol for real-time sensor network applications," in Proc. IEEE International Conference on Communications (ICC 2010), May 2010, pp. 1–6.

- [19] D. Krüandger, D. Pfisterer, and S. Fischer, “CUPID – communication pattern informed duty cycling in sensor networks,” in Proc. Fifth International Conference on Systems and Networks Communications (ICSNC 2010), August 2010, pp. 70–75.
- [20] J. Pollastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in Proc. 2nd ACM International Conference CARRANO et al.: SURVEY AND TAXONOMY OF DUTY CYCLING MECHANISMS IN WIRELESS SENSOR NETWORKS 193 on Embedded Networked Sensor Systems (SenSys '04). New York, NY, USA: ACM, 2004, pp. 95–107.
- [21] A. El-Hoiydi and J.-D. Decotignie, “WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks,” in Algorithmic Aspects of Wireless Sensor Networks - Lecture Notes in Computer Science, S. Nikolettseas and J. Rolim, Eds. Springer Berlin / Heidelberg, 2004, vol. 3121, pp. 18–31.
- [22] TinyOS. (2012, February) TinyOS wiki page on low power implementation over the CC2420 radio. [Online]. Available: <http://docs.tinyos.net/index.php/CC2420> Asynchronous Low Power Listening Implementation.
- [23] M. Buettner, G. V. Yee, E. Anderson, and R. Han, “X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in Proc. 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys '06). New York, NY, USA: ACM, 2006, pp. 307–320.
- [24] Y. Sun, O. Gurewitz, and D. B. Johnson, “RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks,” in Proc. 6th ACM International Conference on Embedded Networked Sensor Systems (SenSys '08). New York, NY, USA: ACM, 2008, pp. 1–14.
- [25] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, “Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks,” in Proc. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), vol. 1, 2002, pp. 200–209.

-
- [26] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, “Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks,” *Mobile Networks and Applications*, vol. 10, pp. 169–181, 2005.
- [27] R. Zheng, J. C. Hou, and L. Sha, “Asynchronous wakeup for ad hoc networks,” in *Proc. 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '03)*. New York, NY, USA: ACM, 2003, pp. 35–45.
- [28] M. Maekawa, “A N algorithm for mutual exclusion in decentralized systems,” *ACM Trans. Comput. Syst.*, vol. 3, pp. 145–159, May 1985.
- [29] García de Jalón, J., Rodríguez, J.I., Brazález, A. (2001). *Aprenda Matlab 6.1 como si estuviera en primero*. ETSII, Universidad Politécnica de Madrid, 113 pp.

GLOSARIO

RSI	Redes de Sensores Inalámbricos.
MAC	Control de acceso al medio. <i>Medium Access Control.</i>
CSMA	Acceso múltiple por detección de portadora. <i>Carrier Sense Multiple Access.</i>
TDMA	Acceso múltiple por división del tiempo. <i>Time Division Multiple Access.</i>
LPL	Escucha de baja potencia. <i>Low Power Listening.</i>
B-MAC	Protocolo de control de acceso al medio Berkeley. <i>Berkeley Medium Access Control.</i>
WiseMAC	Protocolo de control de acceso al medio para sensores inalámbricos. <i>Wireless Sensor MAC.</i>
LPP	Muestreo de baja potencia. <i>Low Power Probing.</i>
D-MAC	Protocolo de control de acceso al medio dinámico. <i>Dynamic MAC.</i>
PELLMAC	Protocolo de control de acceso al medio de consumo eficiente de potencia y baja

latencia. *Power efficient and low latency MAC.*

LEEMAC

Protocolo de control de acceso al medio con eficiente consumo de energía y latencia. *Latency and energy efficient MAC.*

SPEED-MAC

Protocolo de control de acceso al medio de gran velocidad y eficiencia energética. *Speedy and energy efficient data delivery MAC.*

CUPID

Protocolo de control de acceso al medio con información de ciclo de trabajo de radio para la comunicación. *Communication pattern informed duty cycling*

ANEXOS

Anexo I Función para la implementación del mecanismo de ciclo de trabajo de radio: muestreo de preámbulo largo.

`function` N = preamble_sampling(n,it,tw,ts,t)

`%` Eventos

`%` 1 Rx Se despierta

`%` 2 Rx Termina Escucha

`%` 3 Tx Inicia Tx

`%` 4 Tx Termina Tx

`%` Estados de los nodos

`%` 1 Dormido

`%` 2 Despierto

`%` 3 Transmitiendo

`%` 4 Recibiendo

`%` Matriz N

`%` 1 Estado de los nodos

`%` 2 Nodo que recibe (target)

`%` 3 Tiempo en que se inicia el estado actual

`%` 4 Tiempo que se mantuvo dormido

`%` 5 Tiempo que se mantuvo despierto

`%` 6 Tiempo que se mantuvo transmitiendo

```
% 7 Tiempo que se mantuvo recibiendo
% 8 Número de paquetes transmitidos
% 9 Tiempo de inicio de la transmisión
% 10 Tiempo total de la transmisión de paquetes exitosa

pr = 10;
td = 5;
N = zeros(10,n);
N(1,:) = N(1,:) + 1;
EV = zeros(3,n*2);

for i=1:n
    EV(1,i) = rand(1)*(ts+tw);% el nodo se despierta
    EV(2,i) = 1;% primer evento (el Rx se despierta)
    EV(3,i) = i;% nodo que se encuentra en ese estado
    EV(1,n+i) = exprnd(it);% tiempo en que comienza la transmisión
    EV(2,n+i) = 3;% tercer evento (Tx inicia la transmisión)
    EV(3,n+i) = i;% nodo que se encuentra en ese estado
end

time = 0;

while time < t
    [time,i] = min(EV(1,:));% toma el valor mínimo de los tiempos establecidos (guarda la
    posición en i y el tiempo en time
    switch EV(2,i)% cambia de evento según los diferentes casos que se dan a continuación
```

```

case 1% Rx se despierta

    if N(1,EV(3,i)) == 1% Si el nodo está dormido

        if ~(N(2,:) == EV(3,i))% Si no le están transmitiendo

            N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
N(3,EV(3,i));% tiempo que demora en el estado anterior

            N(1,EV(3,i)) = 2;% El nodo se despierta

            N(3,EV(3,i)) = time;% tiempo en que se despierta

            EV(:,i) = [time+tw;2;EV(3,i)];% tiempo en que el receptor termina de escuchar

        else

            N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
N(3,EV(3,i));% tiempo que demora en el estado anterior

            N(1,EV(3,i)) = 4;% el nodo comienza a recibir

            N(3,EV(3,i)) = time;% tiempo en que recibe

        end

    else

        EV(:,i) = [];% borra el evento si no está en las situaciones anteriores

    end

case 2% El receptor termina la escucha

    if N(1,EV(3,i)) == 2% Si el nodo está despierto

        N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
N(3,EV(3,i));% tiempo de duración en ese estado

        N(1,EV(3,i)) = 1;% el nodo vuelve al estado de dormido

        N(3,EV(3,i)) = time;% tiempo que comienza este estado

        EV(:,i) = [time+ts;1;EV(3,i)];% Tiempo en que volverá a despertar

    else

```

```

    EV(:,i) = [];% Borra los eventos que ocurren si no están en la situación anterior
end

case 3% Transmisor inicia la transmisión

if ~(N(1,:) == 3)% Si el nodo no está transmitiendo

    N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
N(3,EV(3,i));% guardo el tiempo que se mantuvo en este estado

    N(1,EV(3,i)) = 3;% Comienza a transmitir

    N(3,EV(3,i)) = time;% Guardo el tiempo en que entra a este estado

    EV(:,i) = [time+pr+td;4;EV(3,i)];% Tiempo en que termina la transmisión

    nodes = 1:n;% variable que contiene los diferentes valores que pueden tener los
nodos

    nodes(nodes == EV(3,i)) = [];% nodes no devuelve nada en caso de que sea el
nodo en cuestión

    target = nodes(randi(n-1,1));% el nodo receptor será un entero aleatorio diferente
de cero

    N(2,EV(3,i)) = target;% nodo que se transmite corresponde con la fila 2 de la
matriz N

    if N(1,target) == 2% Si el nodo que va a recibir está despierto

        N(1,target) = 4;% comienza a recibir

        N(5,target) = N(5,target) + time - N(3,target);% guardo el tiempo en que
permanece en este estado (despierto)

        N(3,target) = time;% guardo el tiempo en que comienza a recibir

    end

else

    EV(1,i) = time+0.1;% Próximo tiempo en que volverá a intentar transmitir

end

```



```

if N(9,EV(3,i)) == 0
    N(9,EV(3,i)) = time;% La fila 9 de la matriz N contiene el tiempo de inicio de la
transmisión
end
case 4% Termina el transmisor de transmitir
    N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time - N(3,EV(3,i));%
Guardo el tiempo que permaneció en este estado(transmitiendo)
    N(10,EV(3,i)) = N(10,EV(3,i)) + time - N(9,EV(3,i));% duración de la transmisión
del paquete exitosa
    N(9,EV(3,i)) = 0;
    N(1,EV(3,i)) = 1;% el nodo vuelve al estado de sueño
    N(3,EV(3,i)) = time;%Guardo el tiempo de terminar la transmisión
    N(8,EV(3,i)) = N(8,EV(3,i)) + 1;% Número de paquetes transmitidos
    target = N(2,EV(3,i));% Nodo que recibe
    N(3+ N(1,target),target) = N((3+ N(1,target)),target) + time - N(3,target);% Guardo
el tiempo en que permanece en este estado(recibiendo)
    N(1,target) = 1;% El nodo vuelve a dormir
    N(3,target) = time;%Guardo el tiempo de terminar la transmisión
    EV = [EV [time+ts;1;target]]; % Próximo tiempo en que despertará el nodo receptor
    N(2,EV(3,i)) = 0;% Ningún nodo está recibiendo
    EV(1,i) = time+exprnd(it);% Tiempo de la próxima transmisión
    EV(2,i) = 3;% El transmisor inicia la transmisión
    EV = [EV [time+ts;1;EV(3,i)]]; % Próximo tiempo en que despertará el nodo
receptor
end

```

end

Anexo II Función para la implementación del mecanismo de ciclo de trabajo de radio: muestreo de preámbulo corto.

function N = int_preamble_sampling(n,it,tw,ts,t)

% Eventos

% 1 Rx Se despierta

% 2 Rx Termina Escucha

% 3 Tx Inicia Tx

% 4 Tx Termina Tx

% Estados de los nodos

% 1 Dormido

% 2 Despierto

% 3 Transmitiendo

% 4 Recibiendo

% Matriz N

% 1 Estado de los nodos

% 2 Nodo que recibe (target)

% 3 Tiempo en que se inicia el estado actual

% 4 Tiempo que se mantuvo dormido

```
% 5 Tiempo que se mantuvo despierto
% 6 Tiempo que se mantuvo transmitiendo
% 7 Tiempo que se mantuvo recibiendo
% 8 Número de paquetes transmitidos
% 9 Tiempo de inicio de la transmisión
% 10 Tiempo total de la transmisión de paquetes exitosa

pr = 10;
td = 5;

N = zeros(10,n);
N(1,:) = N(1,:) + 1;
EV = zeros(3,n*2);

for i=1:n
    EV(1,i) = rand(1)*(ts+tw);% el nodo se despierta
    EV(2,i) = 1;% primer evento (el Rx se despierta)
    EV(3,i) = i;% nodo que se encuentra en ese estado
    EV(1,n+i) = exprnd(it);% tiempo en que comienza la transmisión
    EV(2,n+i) = 3;% tercer evento (Tx inicia la transmisión)
    EV(3,n+i) = i;% nodo que se encuentra en ese estado
end

time = 0;
```

```

while time < t

    [time,i] = min(EV(1,:));% toma el valor mínimo de los tiempos establecidos (guarda la
    posición en i y el tiempo en time

    switch EV(2,i)% cambia de evento según los diferentes casos que se dan a continuación

        case 1% Rx se despierta

            if N(1,EV(3,i)) == 1% Si el nodo está dormido

                if ~(N(2,:) == EV(3,i))% Si no le están transmitiendo

                    N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
                    N(3,EV(3,i));% tiempo que demora en este estado

                    N(1,EV(3,i)) = 2;% El nodo se despierta

                    N(3,EV(3,i)) = time;% tiempo en que se despierta

                    EV(:,i) = [time+tw;2;EV(3,i)];% tiempo en que el receptor termina de escuchar

                else

                    N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
                    N(3,EV(3,i));% tiempo que demora en este estado

                    N(1,EV(3,i)) = 4;% el nodo pasa al estado recibiendo

                    N(3,EV(3,i)) = time;% tiempo en que recibe

                    EV(:,i) = [time+td;4;find(N(2,:) == EV(3,i))];

                end

            else

                EV(:,i) = [];% borra el evento si no está en las situaciones anteriores

            end

        case 2% El receptor termina la escucha

            if N(1,EV(3,i)) == 2% Si el nodo está despierto

```

```

N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
N(3,EV(3,i));% tiempo de duración en ese estado

N(1,EV(3,i)) = 1;% el nodo vuelve al estado de dormido

N(3,EV(3,i)) = time;% tiempo que comienza este estado

EV(:,i) = [time+ts;1;EV(3,i)];% Tiempo en que volverá a despertar

else

EV(:,i) = [];% Borra los eventos que ocurren si no están en la situación anterior

end

case 3% Transmisor inicia la transmisión

if ~(N(1,:) == 3)% Si el nodo no está transmitiendo

N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
N(3,EV(3,i));% guardo el tiempo que se mantuvo en este estado

N(1,EV(3,i)) = 3;% Comienza a transmitir

N(3,EV(3,i)) = time;% Guardo el tiempo en que entra a este estado

%EV(:,i) = [time+pr+td;4;EV(3,i)];% Tiempo en que termina la transmisión

nodos = 1:n;% variable que contiene los diferentes valores que pueden tener los
nodos

nodos(nodos == EV(3,i)) = [];% nodes no devuelve nada en caso de que sea el
nodo en cuestión

target = nodes(randi(n-1,1));% el nodo receptor será un entero aleatorio diferente
de cero

N(2,EV(3,i)) = target;% nodo que se transmite corresponde con la fila 2 de la
matriz N

if N(1,target) == 2% Si el nodo que va a recibir está despierto

N(1,target) = 4;% comienza a recibir

```

```

    N(5,target) = N(5,target) + time - N(3,target);% guardo el tiempo en que
permanece en este estado (despierto)

    N(3,target) = time;% guardo el tiempo en que comienza a recibir

    EV(:,i) = [time+td;4;EV(3,i)];

end

else

    EV(1,i) = time+0.1; %Próximo tiempo en que volverá a intentar transmitir

end

if N(9,EV(3,i)) == 0

    N(9,EV(3,i)) = time;% La fila 9 de la matriz N contiene el tiempo de inicio de la
transmisión

end

case 4% Termina el transmisor de transmitir

    N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time - N(3,EV(3,i));%
Guardo el tiempo que permaneció en este estado

    N(10,EV(3,i)) = N(10,EV(3,i)) + time - N(9,EV(3,i));% duración de la transmisión
del paquete

    N(9,EV(3,i)) = 0;

    N(1,EV(3,i)) = 1;% el nodo está en el estado de sueño

    N(3,EV(3,i)) = time;% Guardo el tiempo de terminar la transmisión

    N(8,EV(3,i)) = N(8,EV(3,i)) + 1;% Número de paquetes transmitidos

    target = N(2,EV(3,i));% Nodo que recibe

    N(3+ N(1,target),target) = N((3+ N(1,target)),target) + time - N(3,target);% Guardo
el tiempo en que permanece en este estado(recibiendo)

```

```

N(1,target) = 1;% El nodo vuelve a dormir
N(3,target) = time;%Guardo el tiempo de terminar la transmisión
EV = [EV [time+ts;1;target]];% Próximo tiempo en que despertará el nodo receptor
N(2,EV(3,i)) = 0;% Ningún nodo está recibiendo
EV(1,i) = time+exprnd(it);% Tiempo de la próxima transmisión
EV(2,i) = 3;% El transmisor inicia la transmisión
EV = [EV [time+ts;1;EV(3,i)]];% Próximo tiempo en que despertará el nodo
transmisor
end
end

```

Anexo III Función para la implementación del mecanismo de ciclo de trabajo de radio: iniciado en el receptor.

```
function N = inic_rx(n,it,tw,ts,t)
```

```

% Eventos
% 1 Rx Se despierta
% 2 Rx Termina Escucha
% 3 Tx Inicia Tx
% 4 Tx Termina Tx

% Estados de los nodos
% 1 Dormido
% 2 Despierto
% 3 Transmitiendo

```

```
% 4 Recibiendo

% Matriz N

% 1 Estado de los nodos
% 2 Nodo que recibe (target)
% 3 Tiempo en que se inicia el estado actual
% 4 Tiempo que se mantuvo dormido
% 5 Tiempo que se mantuvo despierto
% 6 Tiempo que se mantuvo transmitiendo
% 7 Tiempo que se mantuvo recibiendo
% 8 Número de paquetes transmitidos
% 9 Tiempo de inicio de la transmisión
% 10 Tiempo total de la transmisión de paquetes exitosa

td = 5;
N = zeros(10,n);
N(1,:) = N(1,:) + 1;
EV = zeros(3,n*2);

for i=1:n
    EV(1,i) = rand(1)*(ts+tw);% el nodo se despierta
    EV(2,i) = 1;% primer evento (el Rx se despierta)
    EV(3,i) = i;% nodo que se encuentra en ese estado
    EV(1,n+i) = exprnd(it);% tiempo en que comienza la transmisión
```



```

EV(2,n+i) = 3;% tercer evento (Tx inicia la transmisión)

EV(3,n+i) = i;% nodo que se encuentra en ese estado

end

time = 0;

while time < t

    [time,i] = min(EV(1,:));% toma el valor mínimo de los tiempos establecidos (guarda la
    posición en i y el tiempo en time)

    if isempty(EV(2,i))

        z = 1;

    end

    switch EV(2,i)% cambia de evento según los diferentes casos que se dan a continuación

        case 1% Rx se despierta

            if N(1,EV(3,i)) == 1% Si el nodo está dormido

                if ~(N(1,:) == 3) % Si nadie está transmitiendo

                    N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
                    N(3,EV(3,i));% tiempo que demora en este estado(dormido)

                    if ~(N(2,:) == EV(3,i))

                        N(1,EV(3,i)) = 2; %El nodo se despierta

                        EV(:,i) = [time+tw;2;EV(3,i)];%tiempo en que el receptor termina de
                        escuchar

                    else

                        N(1,EV(3,i)) = 4;% El nodo se despierta y comienza a recibir

                        N(3,EV(3,i)) = time;% tiempo que comienza a recibir

```

```

N(1,find(N(2,:) == EV(3,i),1)) = 3;

EV(:,i) = [time+td;4;find(N(2,:) == EV(3,i),1)];%tiempo en que termina la
transmisión

end

else

N(1,EV(3,i)) = 2; %El nodo se despierta

EV(:,i) = [time+tw;2;EV(3,i)];%tiempo en que el receptor termina de escuchar

end

else

if (N(1,EV(3,i)) == 3) || (N(1,EV(3,i)) == 5)

EV(:,i) = [time+ts;1;EV(3,i)]; % Tiempo en que volverá a despertar

else

EV(:,i) = [];

end

end

case 2% El receptor termina la escucha

if N(1,EV(3,i)) == 2% Si el nodo está despierto

N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
N(3,EV(3,i));% tiempo de duración en ese estado(despierto)

N(1,EV(3,i)) = 1;% el nodo vuelve al estado de dormido

N(3,EV(3,i)) = time;% tiempo que comienza este estado

EV(:,i) = [time+ts;1;EV(3,i)];% Tiempo en que volverá a despertar

else

EV(:,i) = [];% Borra los eventos que ocurren si no están en la situación anterior

end

```

```

case 3% Transmisor inicia la transmisión
    if ~(N(1,:) == 3)% Si el nodo no está transmitiendo
        if ~(N(1,EV(3,i)) == 5)% si no está esperando para transmitir
            N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time -
N(3,EV(3,i));% guardo el tiempo que se mantuvo en este estado

            nodes = 1:n;% variable que contiene los diferentes valores que pueden tener
los nodos

            nodes(nodes == EV(3,i)) = [];% nodes no devuelve nada en caso de que sea el
nodo en cuestión

            target = nodes(randi(n-1,1));% el nodo receptor será un entero aleatorio
diferente de cero

            N(2,EV(3,i)) = target;% nodo que se transmite corresponde con la fila 2 de la
matriz N

            N(1,EV(3,i)) = 5;% el nodo espera el beacon del receptor para transmitir

            N(9,EV(3,i)) = time;% La fila 9 de la matriz N contiene el tiempo de inicio de
la transmisión

            EV(:,i) = [];

        else

            EV(1,i) = time+0.1;% Próximo tiempo en que volverá a intentar transmitir

        end

    else

        EV(1,i) = time+0.1;% Próximo tiempo en que volverá a intentar transmitir
    end

```

```

end

case 4% Termina el transmisor de transmitir

    N(3+ N(1,EV(3,i)),EV(3,i)) = N(3+ N(1,EV(3,i)),EV(3,i)) + time - N(3,EV(3,i));%
Guardo el tiempo que permaneció transmitiendo

    N(10,EV(3,i)) = N(10,EV(3,i)) + time - N(9,EV(3,i));% duración de la transmisión
del paquete exitosa

    N(9,EV(3,i)) = 0;

    N(1,EV(3,i)) = 1;% el nodo transmisor regresa al estado de sueño

    N(3,EV(3,i)) = time;% Guardo el tiempo de terminar la transmisión

    N(8,EV(3,i)) = N(8,EV(3,i)) + 1;% Número de paquetes transmitidos

    target = N(2,EV(3,i));% Nodo que recibe

    N(3+ N(1,target),target) = N((3+ N(1,target)),target) + time - N(3,target);% Guardo
el tiempo en que permanece recibiendo

    N(1,target) = 1;% El nodo receptor vuelve a dormir

    N(3,target) = time;% Guardo el tiempo de terminar la transmisión

    EV = [EV [time+ts;1;target]]; % Próximo tiempo en que despertará el nodo receptor

    N(2,EV(3,i)) = 0;% Ningún nodo está recibiendo

    EV(1,i) = time+expnd(it);% Tiempo de la próxima transmisión

    EV(2,i) = 3;% El transmisor inicia la transmisión

    EV = [EV [time+ts;1;EV(3,i)]]; % Próximo tiempo en que despertará el nodo
transmisor

end

end

```

Anexo IV Experimento para analizar el comportamiento del consumo energético en función de la variación del número de nodos en la red.

```
function experimento1(n)
```

```
% Energía consumida VS Número de nodos tomando como referencia el nodo sensor
```

```
Imote2
```

```
D = zeros(1,n-1);
```

```
DD=0;
```

```
for j=1:5
```

```
    for i = 2:n
```

```
        N = preamble_sampling(i,60,0.1,9.9,3000);% valores de entrada a la función variando el número de nodos
```

```
        D(i-1) = mean(N(4,:)).*((20*10-6)2) + mean(N(5,:)).*((0.426*10-3)2) +  
mean(N(6,:)).*((15*10-3)2) + mean(N(7,:)).*((15*10-3)2);% Promedio del tiempo que  
se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia consumida
```

```
    end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5
```

```
save con1.mat
```

```
figure;hold on; plot(2:n,DD,'r')
```

```
DD=0;
```

```
for j = 1:5
```

```
    for i = 2:n
```

```
N = int_preamble_samplng(i,60,0.1,9.9,3000); );% valores de entrada a la función  
variando el número de nodos
```

```
D(i-1) = mean(N(4,:)).*((20*10^-6)^2) + mean(N(5,:)).*((0.426*10^-3)^2) +  
mean(N(6,:)).*((15*10^-3)^2) + mean(N(7,:)).*((15*10^-3)^2);% Promedio del tiempo que  
se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia consumida  
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5
```

```
save con2.mat
```

```
plot(2:n,DD)
```

```
DD=0;
```

```
for j=1:5
```

```
for i = 2:n
```

```
N = inic_rx(i,60,0.1,9.9,3000); );% valores de entrada a la función variando el número de  
nodos
```

```
D(i-1) = mean(N(4,:)).*((20*10^-6)^2) + mean(N(5,:)).*((0.426*10^-3)^2) +  
mean(N(6,:)).*((15*10^-3)^2) + mean(N(7,:)).*((15*10^-3)^2);% Promedio del tiempo que  
se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia consumida  
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5
```

```
save con3.mat
```

```
plot(2:n,DD,'g')
```

Anexo V Experimento para analizar el comportamiento la latencia en la transmisión de paquetes en función de la variación del número de nodos en la red.

```
function experimento1(n)
```

```
% latencia en la transmisión de paquetes VS Número de nodos
```

```
D = zeros(1,n-1);
```

```
DD = 0;
```

```
for j = 1:5
```

```
for i = 2:n
```

```
    N = preamble_sampling(i,60,0.1,9.9,3000); );% valores de entrada a la función variando  
    el número de nodos
```

```
    A= N(10,:)./N(8,:);
```

```
    A(isnan(A))=[];
```

```
    D(i-1) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete
```

```
end
```

```
DD = DD + D;
```

```
end
```

```
DD = DD/5;
```

```
save lat1.mat
```

```
figure;hold on; plot(2:n,DD,'r')
```

```
DD = 0;
```

```
for j = 1:5
for i = 2:n
    N = int_preamble_sampling(i,60,0.1,9.9,3000); );% valores de entrada a la función
    variando el número de nodos

    A= N(10,:)./N(8,:);
    A(isnan(A))=[];
    D(i-1) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete
end
DD = DD + D;
end
DD = DD/5;
save lat2.mat
plot(2:n,DD)

DD = 0;
for j = 1:5
for i = 2:n
    N = inic_rx(i,60,0.1,9.9,3000); );% valores de entrada a la función variando el número de
    nodos

    A= N(10,:)./N(8,:);
    A(isnan(A))=[];
    D(i-1) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete
end
```



```
DD = DD + D;
```

```
end
```

```
DD = DD/5;
```

```
save lat3.mat
```

```
plot(2:n,DD,'g')
```

Anexo VI Experimento para analizar el comportamiento de la tasa de transmisión en función de la variación del número de nodos en la red.

```
function experimento3(n)
```

```
% Tasa de transmisión VS Número de nodos
```

```
D = zeros(1,n-1);
```

```
DD=0;
```

```
for j=1:5
```

```
for i = 2:n
```

```
    N = preamble_sampling(i,60,0.1,9.9,10000); );% valores de entrada a la función  
    variando el número de nodos
```

```
    D(i) = mean(N(8,:))/10000;% Promedio del número de paquetes transmitidos en el  
    tiempo de simulación
```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save tasa1.mat
```

```
figure;hold on; plot(2:n,DD,'r')
```

```
DD=0;
for j=1:5
for i = 2:n
    N = int_preamble_sampling(i,60,0.1,9.9,10000); );% valores de entrada a la función
    % variando el número de nodos
    D(i) = mean(N(8,:))/10000;% Promedio del número de paquetes transmitidos en el
    % tiempo de simulación
end
DD=DD+D;
end
DD=DD/5;
save tasa2.mat
plot(2:n,DD)

DD=0;
for j=1:5
for i = 2:n
    N = inic_rx(i,60,0.1,9.9,10000);
    D(i-1) = mean(N(8,:))/10000;% Promedio del número de paquetes transmitidos en el
    % tiempo de simulación
end
DD=DD+D;
end
DD=DD/5;
save tasa3.mat
```

```
plot(2:n,DD,'g')
```

Anexo VII Experimento para analizar el comportamiento del consumo energético en función de la variación del tráfico.

```
function experimento01
```

```
% Energía consumida VS tráfico
```

```
D = zeros(1,9);
```

```
DD=0;
```

```
for n= 1:5
```

```
    for i = 1:9
```

```
        N = preamble_sampling(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la  
función variando el tráfico
```

```
        D(i) = mean(N(4,:)).*((20*10^-6)^2) + mean(N(5,:)).*((0.426*10^-3)^2) +  
mean(N(6,:)).*((15*10^-3)^2) + mean(N(7,:)).*((15*10^-3)^2);% Promedio del tiempo que  
se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia consumida
```

```
    end
```

```
    DD = DD+D;
```

```
end
```

```
DD = DD/5;
```

```
save con01.mat
```

```
figure;hold on; plot(60:-5:20,DD,'r')
```

```
DD=0;
```

```
for n=1:5
```

```
    for i = 1:9
```

```
N = int_preamble_sampling(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la
función variando el tráfico
```

```
D(i) = mean(N(4,:)).*((20*10^-6)^2) + mean(N(5,:)).*((0.426*10^-3)^2) +
mean(N(6,:)).*((15*10^-3)^2) + mean(N(7,:)).*((15*10^-3)^2); );% Promedio del tiempo
que se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia
consumida
```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save con02.mat
```

```
plot(60:-5:20,DD)
```

```
DD=0;
```

```
for j=1:5
```

```
for i = 1:9
```

```
N = inic_rx(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la función variando
el tráfico
```

```
D(i) = mean(N(4,:)).*((20*10^-6)^2) + mean(N(5,:)).*((0.426*10^-3)^2) +
mean(N(6,:)).*((15*10^-3)^2) + mean(N(7,:)).*((15*10^-3)^2);% Promedio del tiempo que
se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia consumida
```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save con03.mat
```

```
plot(60:-5:20,DD,'g')
```

Anexo VIII Experimento para analizar el comportamiento de la latencia en la transmisión de paquetes en función de la variación del tráfico.

```
function experimento02
```

```
% latencia en la transmisión de paquetes VS tráfico
```

```
D = zeros(1,9);
```

```
DD=0;
```

```
for r = 1:5
```

```
    for i = 1:9
```

```
        N = preamble_sampling(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la  
función variando el tráfico
```

```
        A= N(10,:)./N(8,:);
```

```
        A(isnan(A))=[];
```

```
        D(i) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete
```

```
    end
```

```
    DD=DD+D;
```

```
end
```

```
A=DD/5;
```

```
save lat01.mat
```

```
figure;hold on; plot(60:-5:20,A, 'r')
```

```
DD=0;
```

```
for r = 1:5
```

```
for i = 1:9

    N = int_preamble_sampling(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la
función variando el tráfico

    A= N(10,:)./N(8,:);

    A(isnan(A))=[];

    D(i) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete

end

DD=DD+D;

end

B = DD/5;

save lat02.mat

plot(60:-5:20,B)

DD=0;

for r = 1:5

    for i = 1:9

        N = inic_rx(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la función variando
el tráfico

        A= N(10,:)./N(8,:);

        A(isnan(A))=[];

        D(i) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete

    end

    DD=DD+D;

end

C = DD/5;
```

```
save lat03.mat
```

```
plot(60:-5:20,C, 'g')
```

Anexo IX Experimento para analizar el comportamiento de la tasa de transmisión en función de la variación del tráfico.

```
function experimento03
```

```
% Tasa de transmisión VS tráfico
```

```
D = zeros(1,9);
```

```
for r = 1:5
```

```
    for i = 1:9
```

```
        N = preamble_sampling(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la  
función variando el tráfico
```

```
        D(i) = mean(N(8,:))./3000;% Promedio del número de paquetes transmitidos en el  
tiempo de simulación
```

```
    end
```

```
    DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save tasa01
```

```
figure;hold on; plot(60:-5:20,DD,'r')
```

```
DD=0;
```

```
for x=1:5
```

```
    for i = 1:9
```

```
        N = int_preamble_sampling(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la  
función variando el tráfico
```

```
D(i) = mean(N(8,:))./3000;% Promedio del número de paquetes transmitidos en el tiempo de simulación
```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save tasa02.mat
```

```
plot(60:-5:20,DD)
```

```
DD=0;
```

```
for x=1:5
```

```
for i = 1:9
```

```
    N = inic_rx(25,60 - (i-1)*5,0.1,9.9,3000); );% valores de entrada a la función variando el tráfico
```

```
    D(i) = mean(N(8,:))./3000;% Promedio del número de paquetes transmitidos en el tiempo de simulación
```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save tasa03.mat
```

```
plot(60:-5:20,DD,'g')
```

Anexo X Experimento para analizar el comportamiento del consumo energético en función de la variación del ciclo de trabajo.

```
function experimento001
```


% Energía consumida VS ciclo de trabajo

```
D = zeros(1,5);
```

```
DD=0;
```

```
for x=1:5
```

```
    for i = 1:5
```

```
        N = preamble_sampling(25,60,0.1*i,10-0.1*i,3000); );% valores de entrada a la  
función variando el ciclo de trabajo
```

```
        D(i) = mean(N(4,:)).*((20*10^-6)^2) + mean(N(5,:)).*((0.426*10^-3)^2) +  
mean(N(6,:)).*((15*10^-3)^2) + mean(N(7,:)).*((15*10^-3)^2);% Promedio del tiempo que  
se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia consumida
```

```
    end
```

```
    DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save con001.mat
```

```
figure;hold on; plot(1:5,DD,'r')
```

```
DD=0;
```

```
for x=1:5
```

```
    for i = 1:5
```

```
        N = int_preamble_sampling(25,60,0.1*i,10-0.1*i,3000); % valores de entrada a la  
función variando el ciclo de trabajo
```

```

D(i) = mean(N(4,:)).*((20*10^-6)^2) + mean(N(5,:)).*((0.426*10^-3)^2) +
mean(N(6,:)).*((15*10^-3)^2) + mean(N(7,:)).*((15*10^-3)^2); );% Promedio del tiempo
que se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia
consumida

```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save con002.mat
```

```
plot(1:5,DD)
```

```
DD=0;
```

```
for x=1:5
```

```
for i = 1:5
```

```

N = inic_rx(25,60,0.1*i,10-0.1*i,3000); % valores de entrada a la función variando el
ciclo de trabajo

```

```

D(i) = mean(N(4,:)).*((20*10^-6)^2) + mean(N(5,:)).*((0.426*10^-3)^2) +
mean(N(6,:)).*((15*10^-3)^2) + mean(N(7,:)).*((15*10^-3)^2); );% Promedio del tiempo
que se mantuvo transmitiendo, recibiendo, en escucha y durmiendo por la potencia
consumida

```

```
end
```

```
DD = DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save con003.mat
```

```
plot(1:5,DD,'g')
```

Anexo XI Experimento para analizar el comportamiento la latencia en la transmisión de paquetes en función de la variación del ciclo de trabajo.

```
function experimento002
```

```
% latencia en la transmisión de paquetes VS ciclo de trabajo
```

```
D = zeros(1,5);
```

```
DD=0;
```

```
for x=1:5
```

```
    for i = 1:5
```

```
        N = preamble_sampling(25,60,0.1*i,10-0.1*i,3000); % valores de entrada a la función  
        variando el ciclo de trabajo
```

```
        A= N(10,:)./N(8,:);
```

```
        A(isnan(A))=[];
```

```
        D(i) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete
```

```
    end
```

```
    DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save lat001.mat
```

```
figure;hold on; plot(1:5,DD,'r')
```

```
DD=0;
```

```
for x=1:5
```

```
    for i = 1:5
```

```
N = int_preamble_sampling(25,60,0.1*i,10-0.1*i,3000); % valores de entrada a la
función variando el ciclo de trabajo
```

```
A= N(10,:)./N(8,:);
```

```
A(isnan(A))=[];
```

```
D(i) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete
```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save lat002.mat
```

```
plot(1:5,DD)
```

```
DD=0;
```

```
for x=1:5
```

```
    for i = 1:5
```

```
        N = inic_rx(25,60,0.1*i,10-0.1*i,3000); % valores de entrada a la función variando el
ciclo de trabajo
```

```
        A= N(10,:)./N(8,:);
```

```
        A(isnan(A))=[];
```

```
        D(i) = mean(A);% Promedio del tiempo que demora el nodo en transmitir un paquete
```

```
    end
```

```
    DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save lat003.mat
```

```
plot(1:5,DD,'g')
```

Anexo XII Experimento para analizar el comportamiento la tasa de transmisión de paquetes en función de la variación del ciclo de trabajo.

```
function experimento003
```

```
% tasa de transmisión VS ciclo de trabajo
```

```
D = zeros(1,5);
```

```
DD=0;
```

```
for x=1:5
```

```
    for i = 1:5
```

```
        N = preamble_sampling(25,60,0.1*i,10-0.1*i,3000); % valores de entrada a la función  
        variando el ciclo de trabajo
```

```
        D(i) = mean(N(8,:))./3000;% Promedio del número de paquetes transmitidos en el  
        tiempo de simulación
```

```
    end
```

```
    DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save tasa001.mat
```

```
figure;hold on; plot(1:5,DD,'r')
```

```
DD=0;
```

```
for x=1:5
```

```
    for i = 1:5
```

```
N = int_preamble_sampling(25,60,0.1*i,10-0.1*i,3000); % valores de entrada a la
función variando el ciclo de trabajo
```

```
D(i) = mean(N(8,:))./3000;% Promedio del número de paquetes transmitidos en el
tiempo de simulación
```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save tasa002.mat
```

```
plot(1:5,DD)
```

```
DD=0;
```

```
for x=1:5
```

```
for i = 1:5
```

```
N = inic_rx(25,60,0.1*i,10-0.1*i,3000); % valores de entrada a la función variando el
ciclo de trabajo
```

```
D(i) = mean(N(8,:))./3000;% Promedio del número de paquetes transmitidos en el
tiempo de simulación
```

```
end
```

```
DD=DD+D;
```

```
end
```

```
DD=DD/5;
```

```
save tasa003.mat
```

```
plot(1:5,DD,'g')
```