



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**DIFUSIÓN DE MENSAJES EN UNA RED
INALÁMBRICA DE SENSORES CON ANTENAS
DIRECCIONALES**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

**LICENCIADA EN CIENCIAS DE LA
COMPUTACIÓN**

P R E S E N T A:

GUADALUPE FABIOLA GARCÍA MELÉNDEZ



**DIRECTOR DE TESIS:
DR. SERGIO RAJSBAUM GORODEZKY
2013**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Hoja de Datos del Jurado

1. Datos del alumno

García Meléndez Guadalupe Fabiola
54 25 57 40
Universidad Nacional Autónoma de México
Facultad de Ciencias
Ciencias de la Computación
304148783

2. Datos del tutor

Dr. Sergio Rajsbaum Gorodezky

3. Datos del sinodal 1

Dr. José David Flores Peñaloza

4. Datos del sinodal 2

Dr. José de Jesús Galaviz Casas

5. Datos del sinodal 3

Dra. María de Luz Gasca Soto

6. Datos del sinodal 4

Dr. Víctor Rangel Licea

7. Datos del trabajo escrito

Difusión de mensajes en una red inalámbrica de sensores con antenas direccionales.

78 páginas

2013

*Dedicado con mucho cariño
a mi familia y a Philipp*

Agradecimientos

En primer lugar agradezco a mi mamá, por quedarse conmigo, por cuidarme y luchar para verme crecer dignamente, ya que sin ella hubiera sido mucho más difícil llegar a este momento. Agradezco enormemente a Arturo por ayudarnos tanto y entregar su vida a nuestra familia, pues debido a esa entrega y apoyo tuve la oportunidad de llegar hasta este punto de mi vida. Me complace agradecer a mis hermosas hermanas, Tania y Karla, por ser tanto un gran ejemplo de vida como una enorme motivación para lograr terminar este trabajo. Agradezco a mi padre, que a pesar de la distancia me apoyó siempre que pudo hacerlo y estuvo al pendiente de mi avance.

También me gustaría agradecer a mi mamá Guille y a mi papá Alfredo, quién seguramente está contento y orgulloso de que estoy a punto de lograrlo. Gracias a ambos por ser como unos padres para mi y creer en mi capacidad para llegar hasta aquí. Gracias a mis primos, Alfredo, Israel y Daniel, por siempre brindarme su ayuda, la cual me permitió continuar en el camino en muchas ocasiones.

Muchísimas gracias a Diana y a Mariana; mis primas, amigas y hermanas. Gracias por echarme porras para seguir adelante. Gracias a mi tío Sergio y a mi tía Paty, por ofrecerme siempre su apoyo.

Agradezco a todos mis primos y tíos que me faltan por mencionar, simplemente son demasiados y no puedo mencionar a cada uno. Sin embargo, estoy muy agradecida con cada uno de ellos por apoyarme en el camino y ser parte de una gran familia.

Quiero agradecer de manera muy especial a mi pareja, Philipp, pues me ha demostrado que no existen imposibles en esta vida. Gracias amor por creer en mí de esa manera.

Agradezco a mi gran amiga Cynthia, por escucharme, alentarme y darme fuerzas con sus palabras. Por regañarme y recordarme que debo creer en mi. Mil gracias a Pamela y Jessica, amigas y compañeras en este camino, por brindarme su amistad y creer en que podría lograrlo.

Agradezco a todos aquellos profesores que han participado en mi formación. Ellos han sido además de mis profesores en clase, mis maestros de la vida. En especial quiero agradecer a Favio, que siempre estuvo al pendiente de mis avances y me motivó a seguir adelante.

Mil gracias a mi asesor Sergio Rajsbaum, por su infinita paciencia y su motivación.

Agradezco a mis sinodales por retroalimentar tanto a este trabajo como a mi. Además, agradezco también al Proyecto PAPIIT IN104711 por su gran apoyo y colaboración para poder llevar a cabo este trabajo.

Índice general

Lista de figuras	II
Introducción	1
1. Preliminares	6
1.1. Red inalámbrica de sensores	6
1.2. Algunas aplicaciones	8
1.2.1. Macroscopio de secuoya	8
1.2.2. Monitoreo de la salud	9
1.2.3. Rastreo de animales	9
1.2.4. Prevención del ataque llamado agujero de gusano (<i>Worm-hole attack</i>)	9
1.3. Modelos	10
1.3.1. Modelo de comunicación	10
1.3.2. Modelo de red	12
2. Problemas básicos en redes inalámbricas	14
2.1. Protocolos de Red Distribuidos	15
2.1.1. Propagación de información	16
2.1.2. Pruebas de conectividad	16
2.1.3. Trayectorias más cortas	18
2.1.4. Cambios topológicos	18
2.2. Algoritmos de inversión de enlaces	19
2.2.1. Ruteo en un sistema distribuido	20
2.2.2. Exclusión mutua en un sistema distribuido	21
2.3. Eficiencia energética en la difusión de mensajes en una red inalámbrica de sensores con antenas direccionales	24
2.4. Descubrimiento de vecinos en una red de sensores con antenas direccionales	25

2.4.1. Algoritmos deterministas	25
2.4.2. Algoritmos de descubrimiento de vecinos aleatorios	26
3. Difusión de mensajes	28
3.1. Planteamiento del problema	28
3.2. Algoritmos para la difusión de mensajes	29
3.2.1. Algoritmo de difusión de mensajes	29
3.2.2. Algoritmo AGDM	32
3.3. Ejemplo de difusión	37
Conclusiones	49
Glosario	53
Apéndices	55
A. Teoría de gráficas	55
B. Teoría de redes	59
Bibliografía	63

Índice de figuras

1.	Comparación del rango de transmisión de una antena direccional y una omnidireccional.	2
1.1.	Rango y área de transmisión de dos antenas u y v	8
1.2.	Ejemplo de la posición de antenas en la que hay comunicación.	11
1.3.	Sector de una antena u con ángulo ϕ_u y radio R	11
1.4.	Sectores numerados de un nodo sensor u	11
1.5.	Representación de la transmisión de la antena del nodo v_m sobre su sector i	12
1.6.	Posición de vértices en una gráfica para establecer comunicación	12
1.7.	Ejemplo de notación	13
3.1.	Algoritmo para la Difusión de Mensajes (ADM)	30
3.2.	Gráfica G y su árbol de difusión	31
3.3.	Algoritmo Generador del Árbol de Difusión de Mensajes (AGDM)	33
3.4.	Consumo de 1500 unidades de energía para enviar un mensaje de v_1 a su vecinos en el sector tres	34
3.5.	Ejemplo del inicio de difusión de un mensaje	38
3.6.	Conjuntos disponibles en la primera iteración	38
3.7.	Estado del árbol antes de segunda iteración	39
3.8.	Conjuntos disponibles en la segunda iteración	40
3.9.	Estado del árbol antes de tercera iteración	41
3.10.	Conjuntos disponibles en la tercera iteración	42
3.11.	Estado del árbol antes de la cuarta iteración	43
3.12.	Conjuntos disponibles en la cuarta iteración	44
3.13.	Estado del árbol antes de la quinta iteración	45
3.14.	Conjuntos disponibles en la quinta iteración	45
3.15.	Árbol de difusión	46

3.16. Ejemplo de la rotación de antenas en ADM	48
3.17. Ejemplo del árbol de difusión con tiempos de transmisión . . .	51
3.18. Ejemplo del árbol de difusión con diferente topología	52
A.1. Ejemplo de una gráfica G	56
A.2. Ejemplo de un camino en una gráfica	56
A.3. Ejemplo de una gráfica conexa y una no conexa	57
A.4. Coloración de una gráfica	57
A.5. Ejemplo de una gráfica dirigida	57
A.6. Ejemplo de un árbol	58
B.1. Red de sensores	60
B.2. Espectro electromagnético	61
B.3. Ejemplos de topologías de red	62

Introducción

Antecedentes

Una red inalámbrica de sensores es una red de comunicación que puede ser utilizada para monitorear condiciones físicas o ambientales como: temperatura, sonido, vibración, presión y contaminación. Además de ser útil en la detección sísmica, vigilancia militar, monitoreo de hábitat, monitoreo remoto de pacientes, entre otros.

En muchos casos, el ambiente monitoreado no tiene una infraestructura para energía, así que se vuelve imprescindible para los nodos sensores sobrevivir con fuentes pequeñas y finitas. En consecuencia, el consumo de energía es el factor más importante para determinar la vida de la red.

La red utiliza señales de radio o infrarrojas provenientes de antenas localizadas en los nodos, para enviar o recibir datos. Dichas señales deben ser lo suficientemente fuertes para ser detectadas e interpretadas propiamente. El tipo de antenas más utilizadas son las antenas omnidireccionales. Sin embargo, las antenas direccionales han demostrado reducir el consumo de energía. Esta ventaja surge por la manera en que ésta se transmite. Con una antena omnidireccional, una pequeña porción de la energía transmitida es interceptada por otra antena en la red y el resto, que es difundido a los alrededores, causa interferencias a otros usuarios. Por otro lado, una antena direccional puede transmitir la misma cantidad de energía a un área específica con rango de transmisión mayor. De manera que, otras antenas pueden llegar a interceptar la energía que no podían cuando era transmitida por una antena omnidireccional. Esto indica que las antenas direccionales brindan un mayor reuso espacial del medio compartido al conectar nodos en la red que normalmente no lo estaban, característica que trae consigo un ahorro energético. La

Figura 1 ilustra las diferencias entre el área y el rango de transmisión de una antena direccional y una omnidireccional.

No obstante, el tiempo necesario para que un nodo encuentre a sus nodos vecinos aumenta, ya que tanto el nodo receptor como el emisor deben alinear sus antenas de forma sincronizada y estar dentro de sus rangos de transmisión para poder llevar a cabo la comunicación. Básicamente, un nodo debe descubrir a sus nodos vecinos lo más pronto posible, lo que se traduce a eficiencia energética. Además de permitir a otros algoritmos comenzar su ejecución.

Algunos beneficios adicionales que las antenas direccionales pueden proporcionar son tanto el aumento de capacidad como la seguridad en una red inalámbrica.

El uso de antenas direccionales conlleva al análisis y diseño de algoritmos que permitan resolver problemas básicos y fundamentales en una red tales como: ruteo y difusión de información.

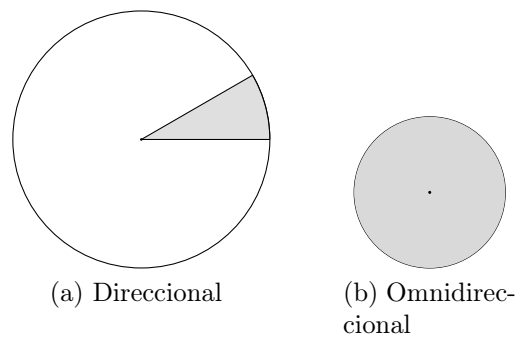


Figura 1: Comparación del rango de transmisión de una antena direccional y una omnidireccional.

Definición del problema

Existen mensajes que necesitan ser enviados a todos los nodos de una red, por lo que es necesario un algoritmo que garantice que estos mensajes serán entregados. Este proceso se denomina *difusión de mensajes* y se define

de la siguiente manera: dada una red inalámbrica de sensores con antenas direccionales y un mensaje M que necesita difundirse, al surgir la petición de difusión de M desde un nodo u en la red, se debe enviar a M a cada uno de los nodos restantes en la red.

Objetivos y Metas

Los objetivos de este texto son que el lector comprenda el por qué de la importancia del ahorro energético en una red inalámbrica de sensores. Además de proponer una solución al problema de difusión de mensajes en este tipo de redes equipadas con antenas direccionales. Para llevar a cabo lo anterior, se definirán dos algoritmos que se basan en algoritmos definidos en los textos escritos por los autores Li et al. [5] para la difusión de mensajes en una red inalámbrica de sensores y por Kranakis et al. [2] para el descubrimiento de vecinos también en una red inalámbrica de sensores. Se modificarán algunos de los algoritmos que estos autores proponen para construir la solución al problema de difusión de mensajes.

Resultados

Se logra comprender por qué resulta importante disminuir el consumo de energía de los procesos que se realizan en una red inalámbrica de sensores. Además, se demuestra teóricamente que los algoritmos propuestos dan una solución para la difusión de mensajes.

Trabajos relacionados

Segall [8] considera la situación donde un número de unidades físicas de cómputo trabajan sobre un mismo problema y en la que su operación es coordinada por medio de enlaces de comunicación que conectan algunas de estas unidades. Se presentan *protocolos de red distribuidos (DNP)* cuyas principales aplicaciones se encuentran en la comunicación de redes. De modo que, algunos problemas para los cuales los DNP han sido propuestos son: ruteo de información, resincronización de redes, búsqueda de rutas más cortas, obtención del árbol de peso mínimo, propagación de información (*broadcast information*), entre otras. El autor propone protocolos para la propagación

de información en una red, para las pruebas de conectividad y para la obtención de las trayectorias con el mínimo número de enlaces desde cualquier nodo a otro.

Welch et al. [14] proponen algoritmos de inversión de enlaces (*Link reversal algorithms*). Estos algoritmos han sido utilizados en varios algoritmos distribuidos para una variedad de problemas. El sistema distribuido es modelado como una gráfica, en la cual los vértices representan las unidades de cómputo y las aristas representan alguna característica del sistema. Se asignan orientaciones virtuales a cada arista de la gráfica para indicar hacia dónde un nodo en la red puede transmitir información. Dicha asignación da como resultado una gráfica dirigida que los autores denominan *orientación* de una gráfica. De este modo, durante la ejecución de un algoritmo, la gráfica puede tener distintas orientaciones. Con este modelo, los autores proporcionan dos algoritmos para el ruteo de información en una gráfica, denominados algoritmo de inversión completa (*full reversal*) y algoritmo de inversión parcial (*Partial reversal*), que indican cómo las aristas deben cambiar su orientación virtual para poder llevar a cabo la transmisión de mensajes de cualquier nodo en la red a un nodo específico, llamado *nodo destino*. Los algoritmos difieren en la elección de las aristas a invertir y en cómo es implementada esta inversión. Se proporcionan también algoritmos para resolver el problema de exclusión mutua en un sistema distribuido.

Chiwewe [1] presenta una técnica de control para topologías distribuidas que mejora la eficiencia energética y reduce las interferencias de radio en redes de sensores inalámbricas. Cada nodo en la red toma decisiones locales acerca de su potencia de transmisión y la culminación de estas decisiones produce una topología de red que preserva la conectividad global. La topología que se produce consiste en una gráfica plana. El tiempo de vida de la red se incrementa mediante la reducción de la potencia de transmisión.

Vasudevan et al. [13] proponen algoritmos probabilísticos para el descubrimiento de vecinos donde los nodos realizan transmisiones independientes y aleatorias para descubrir a sus vecinos. Los algoritmos que proponen se dividen en dos grupos: en uno, los nodos descubren a sus vecinos después de recibir una transmisión de los mismos y, en otro, los nodos dan información sobre su localización para permitir un descubrimiento más rápido.

Pei et al. [6] proponen un incremento gradual del rango de transmisión para propósitos del descubrimiento de vecinos en redes inalámbricas con antenas direccionales. Los vecinos más cercanos se descubren primero y vecinos más alejados se descubren en etapas posteriores.

Organización del trabajo

El texto se organiza de la siguiente manera: en el Capítulo 1 se discuten los preliminares del trabajo, en el cual se muestran algunas aplicaciones de redes inalámbricas de sensores y las ventajas de usar antenas direccionales. Además, se presentan los modelos de comunicación y de red a utilizar.

El Capítulo 2 explica con más detalle algunos de los trabajos que se han realizado sobre redes inalámbricas, con la finalidad de entender de una manera general e intuitiva las soluciones que se proporcionan a aquellos problemas comunes en una red.

El Capítulo 3 analiza la difusión de mensajes en una red inalámbrica de sensores equipados con una antena direccional. En ese capítulo se introducen dos algoritmos para llevar a cabo dicha difusión y se analiza un ejemplo de la ejecución de ambos.

Al final se presentan las conclusiones y se discute sobre trabajo futuro. Además, se proporciona un glosario con los acrónimos que se utilizan a lo largo del texto y dos apéndices que incluyen conceptos básicos tanto de la teoría de gráficas como de la teoría de redes.

Capítulo 1

Preliminares

En este capítulo se define a una red inalámbrica de sensores y se discute sobre las diferencias entre usar una antena direccional y una omnidireccional para llevar a cabo la transmisión inalámbrica. Se mencionan algunas aplicaciones de este tipo de redes y por último se exponen los modelos de comunicación y red que se van a utilizar en el resto del texto.

1.1. Red inalámbrica de sensores

Una red inalámbrica de sensores está compuesta de múltiples estaciones de detección llamadas *nodos sensores*, capaces de llevar a cabo procesamiento local y comunicación inalámbrica. Cada nodo está compuesto por un procesador, memoria, batería, sensores y antenas. En estas redes los dispositivos funcionan con una fuente finita de energía, la cual debe ser manejada de manera óptima para el procesamiento y comunicación.

En muchos casos, los nodos son implantados en ambientes que carecen de infraestructura e incluso a los cuales no se puede tener acceso. Entonces, depende de los nodos identificar su conectividad y su distribución, además de organizarse así mismos y reconfigurarse bajo cualquier cambio en la red. Por ejemplo, al agregar o quitar nodos en la red, los nodos restantes en ella deben poder identificar la situación y seguir trabajando adecuadamente. Una característica importante es el trabajo cooperativo entre los nodos para realizar una tarea específica, pues usan sus capacidades de procesamiento local para transmitir sólo los datos requeridos y parcialmente procesados a otro nodo en la red, disminuyendo la cantidad de información transmitida.

La conectividad en una red inalámbrica de sensores puede ser establecida usando antenas omnidireccionales o direccionales. Las primeras emiten energía uniformemente en todas las direcciones, mientras que las direccionales la emiten en una dirección específica teniendo un rango de transmisión mayor. Las antenas direccionales pueden ser más eficientes y transmitir aún más en una dirección dada por la misma cantidad de energía que las omnidireccionales. Específicamente, el área de cobertura de una antena omnidireccional con rango r es generalmente modelada por un círculo de radio r con un consumo de energía proporcional a $\pi \cdot r^2$. Por otro lado, una antena direccional con propagación angular ϕ y rango R es modelada como un *sector* circular de ángulo ϕ y radio R y que consume una energía proporcional a $\phi \cdot \frac{R^2}{2}$. Por lo tanto, para un costo de energía E , una antena omnidireccional puede alcanzar una distancia de $\sqrt{\frac{E}{\pi}}$ unidades y una direccional alcanza una distancia de $\sqrt{\frac{2E}{\phi}}$ unidades.

En la Figura 1.1 se observa la diferencia entre los rangos y áreas de transmisión de dos antenas u y v , direccional y omnidireccional respectivamente. La energía utilizada por ambas antenas es de 314.16 unidades. Se observa que al usar la antena direccional se tiene un rango de transmisión mayor en comparación con la antena omnidireccional en la que el área de cobertura corresponde a toda el área del círculo, cubriendo a un rango menor. El primer círculo tiene un radio de 3.4 cm y el segundo de 1 cm en una escala 1:10.

Se ha tenido bastante interés en el estudio de las redes inalámbricas de sensores equipadas con antenas direccionales debido a las grandes ventajas que pueden brindar en varios aspectos a una red. Sin embargo, su uso trae consigo cambios en el funcionamiento y estructura de la misma. Aparecen preguntas como: ¿Cómo cambia la topología de las redes cuando se sustituyen a las antenas omnidireccionales con direccionales y qué algoritmos de orientación y comunicación entre los nodos dan mejores resultados? El análisis de estos y otros aspectos son necesarios, así como el diseño de soluciones para los problemas que puedan presentarse, como por ejemplo, asegurar la conectividad de los nodos en la red.

Este tipo de redes tienen distintas aplicaciones tanto en monitoreo como en rastreo. En la parte de monitoreo se encuentra el monitoreo físico y ambiental (temperatura, sonido, vibración, presión y contaminación), de la

salud, de energía, de fábricas y sus procesos de automatización, el monitoreo sísmico y de hábitat (determinando las plantas, así como la población de animales y su conducta). Las aplicaciones de rastreo incluyen el rastreo de objetos, animales, seres humanos y vehículos.

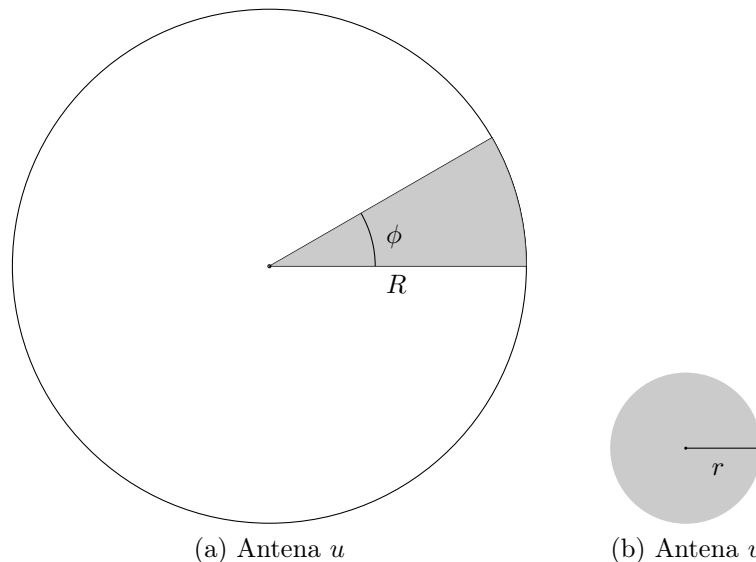


Figura 1.1: Rango y área de transmisión de dos antenas u y v .

1.2. Algunas aplicaciones

En esta sección se explica en que consisten algunas de las aplicaciones de las redes inalámbricas de sensores.

1.2.1. Macroscopio de secuoya

Una red inalámbrica de sensores se encarga de monitorear y grabar a los árboles de secuoya en Sonoma, California [12]. Cada nodo sensor está ubicado a distintas alturas del árbol y miden la temperatura del aire, la humedad y la radiación solar. Los biólogos rastrean los cambios en el clima en los alrededores y pueden validar sus teorías.

1.2.2. Monitoreo de la salud

Usando las redes inalámbricas de sensores se puede mejorar el cuidado y monitorear la salud de un paciente. Algunos diseños se han desarrollado para este tipo de aplicaciones, por ejemplo: el control infantil, el monitoreo y rastreo de la presión arterial, el monitoreo de signos vitales y las alertas para personas con sordera.

Debido a que muchos bebés mueren cada año a causa del Síndrome de Muerte Súbita del Lactante, *Sleep Safe* [16] está diseñado para monitorear a un niño mientras duerme. Se detecta la posición del bebé cuando está durmiendo y se alerta a los padres cuando el niño está acostado sobre su estómago.

FireLine [16] es un sistema de detección de frecuencia cardíaca. Se utiliza para monitorear la frecuencia de bombeo en tiempo real y detectar cualquier anomalía o situación de estrés.

1.2.3. Rastreo de animales

El sistema *ZebraNet* [17] es una red inalámbrica de sensores móvil usada para rastrear migraciones de animales, en este caso, de cebras. Los datos obtenidos son tomados de un GPS y mandados a la estación base a través de otras cebras. Con estos datos, los biólogos pueden entender mejor los movimientos de estos animales durante el día y la noche.

1.2.4. Prevención del ataque llamado agujero de gusano (*Wormhole attack*)

Este tipo de ataques consisten en introducir y conectar dos nodos intermedios dentro de una red inalámbrica [15]. Los agresores pueden construir información de ruteo falsa, borrar paquetes y crear ciclos de ruteo para gastar la energía de la red. Uno de los beneficios al usar antenas direccionales es la obtención de una dirección aproximada de los sensores vecinos y así ayudar a prevenir la intervención de nodos falsos en la red. Entonces, la implementación de protocolos en donde los nodos comparten información de manera direccional ayuda a prevenir el intercambio de información con nodos falsos en la red proporcionando seguridad.

1.3. Modelos

En esta sección se exhibe el modelo de comunicación, presentando el tipo de antenas a utilizar y las condiciones necesarias para que dos nodos puedan comunicarse. Además, se presenta el modelo de la red, indicando la representación de los nodos sensores y sus enlaces de comunicación.

1.3.1. Modelo de comunicación

Varios modelos de comunicación son posibles para un par de nodos sensores con antenas omnidireccionales o direccionales. Considere el par (X, Y) , el cual indica que el nodo sensor emisor utiliza una antena del tipo X para enviar mensajes y el nodo sensor receptor utiliza una antena del tipo Y para recibirlos. En este texto se utiliza un modelo (D, D) , es decir, ambos nodos sensores están equipados con una antena direccional para recibir o enviar mensajes.

Con el modelo (D, D) , tanto el nodo receptor como el emisor deben dirigir sus antenas al mismo tiempo hacia el otro y encontrarse dentro de sus rangos de transmisión para poder tener comunicación. En la Figura 1.2 se muestra la posición que las antenas de dos nodos deben tener para poder comunicarse entre sí. Debido a esto, un nodo puede girar su antena en sentido contrario a las manecillas del reloj y de un sector a otro cuando le sea requerido.

Se supone una *comunicación dúplex*, pues los nodos sensores pueden mandar y recibir mensajes al mismo tiempo ignorando colisiones entre ellos.

Por otra parte, el ángulo de cada sector de un nodo u es $\phi_u = \frac{2\pi}{k_u}$, siendo k_u el entero positivo correspondiente al número de sectores de la antena. En la Figura 1.3 se muestra el ejemplo de un sector con ángulo ϕ_u y radio R . El área de este sector corresponde al área de transmisión de la antena de u . Además, cada sector es disjunto de cualquier otro y está etiquetado con un entero positivo $0 \leq i < k_u$; $k_u = \frac{2\pi}{\phi_u}$. La Figura 1.4 muestra el ejemplo de un nodo sensor que posee 4 sectores y la numeración de ellos.

Cada nodo u en la red tiene k_u variables distintas, denominadas d_{ui} con $i \in \{0, 1, \dots, k_u - 1\}$ y que indican el tiempo que su antena debe mantener la transmisión sobre el sector i . Por ejemplo, si la variable d_{u0} de un nodo

u tiene un valor de 5, la transmisión de energía se mantendrá durante cinco unidades de tiempo sobre el sector cero, siempre que la antena de u gire hacia ese sector.

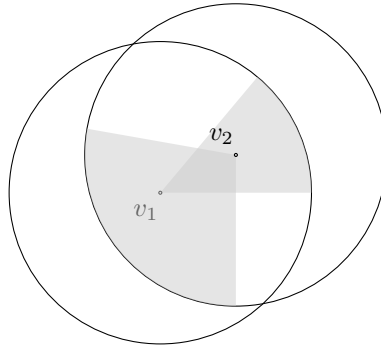


Figura 1.2: Ejemplo de la posición de antenas en la que hay comunicación.

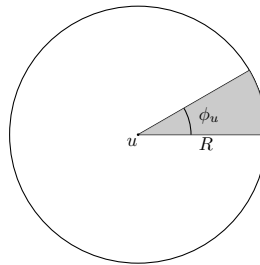


Figura 1.3: Sector de una antena u con ángulo ϕ_u y radio R

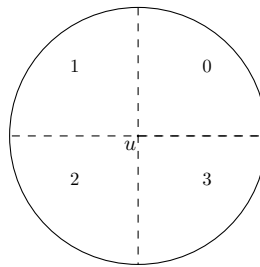


Figura 1.4: Sectores numerados de un nodo sensor u

Otra característica importante es que cada nodo conoce a cada uno de sus nodos vecinos y el sector por el cual puede comunicarse con cada uno de ellos.

1.3.2. Modelo de red

En primer lugar, la red se representa mediante una gráfica $G = (V, E)$ no dirigida y conexa, donde el conjunto de vértices o nodos, $V = \{v_0, v_1, \dots, v_{n-1}\}$, representa los nodos sensores en la red y el conjunto de aristas o enlaces, $E = \{(v_i, v_j) \mid i \neq j; i, j \in \mathbb{N}\}$, representa las conexiones inalámbricas entre los nodos sensores. Así que la arista (v_i, v_j) indica que ambos nodos son *vecinos*, es decir, que cada uno se encuentra dentro del área de transmisión del otro.

Después se obtiene un *árbol* de G que contiene a todos sus vértices y que se denomina *árbol de difusión*. Durante la difusión de un mensaje se asignan orientaciones virtuales a algunas aristas del árbol, las cuales representan hacia qué o de qué vértice se puede enviar o recibir un mensaje en determinado momento. Estas orientaciones producen una versión dirigida del árbol original y pueden cambiar durante la difusión del mensaje.

Una arista (v_m, v_n) , con $m < n$, siempre estará etiquetada con un par (i, j) , que indica que el nodo v_m tiene como vecino a v_n en su sector i y que v_n tiene como vecino a v_m en su sector j . Además, para representar la transmisión de la antena del nodo v_m sobre su sector i , se agrega una orientación a la arista (v_m, v_n) que apunta hacia v_n , tal como se muestra en la Figura 1.5.

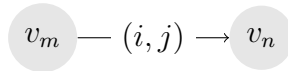


Figura 1.5: Representación de la transmisión de la antena del nodo v_m sobre su sector i

Con esta nueva representación, dos nodos vecinos podrán llevar a cabo la transmisión de un mensaje en determinado momento si cada uno de ellos tiene una flecha apuntando hacia él, pues esto indica que cada uno tiene la posición indicada para establecer la comunicación. En la Figura 1.6 se muestra como se ven en esta representación los nodos de la Figura 1.2.

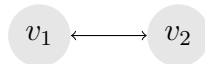


Figura 1.6: Posición de vértices en una gráfica para establecer comunicación

Por otra parte, cada nodo u tiene asociadas dos variables, la primera se denomina A_u e indica el área de transmisión de sus sectores, que es equivalente al consumo de energía de su antena al transmitir por alguno de ellos; la segunda es k_u , que representa el número de sectores de su antena. El conjunto $N_i(u)$ de un nodo u contiene a sus nodos vecinos en el sector i .

En la Figura 1.7 se muestra un ejemplo de los valores que pueden tomar las variables antes mencionadas. La etiqueta $(0, 1)$ de la arista (v_0, v_1) indica que v_0 tiene como vecino a v_1 en su sector cero y que v_1 tiene como vecino a v_0 en su sector uno. Las orientaciones de la arista indican que en ese momento las antenas de ambos nodos están transmitiendo sobre esos sectores y entonces ambos pueden comunicarse.

El área de transmisión de v_1 es de 450 unidades y tiene dos sectores. Además de que el conjunto $N_1(v_1)$ contiene a v_0 y a v_2 .

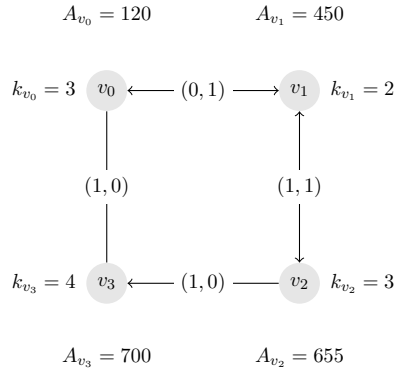


Figura 1.7: Ejemplo de notación

La red se considera una *red síncrona*, es decir, la operación de cada nodo se efectúa en una secuencia de pasos discretos o unidades de tiempo. En una unidad, un nodo primero envía cero o más mensajes, recibe cero o más mensajes y, finalmente, realiza los cálculos locales. De esta manera, los mensajes son recibidos en la misma unidad de tiempo en que fueron enviados. Es decir, si un nodo p envía un mensaje a q en su i -ésima unidad de tiempo, entonces el mensaje es recibido por q en la i -ésima unidad de tiempo, [11].

Capítulo 2

Problemas básicos en redes inalámbricas

En este capítulo se explican los algoritmos propuestos en diversas publicaciones, principalmente artículos. Algunos de estos algoritmos dan solución a problemas comunes en una red, tales como la propagación de información, la búsqueda de las trayectorias más cortas, entre otros; así como a problemas que se presentan en las redes inalámbricas de sensores con antenas direccionales, como el descubrimiento de vecinos.

El capítulo se compone de cinco secciones y en cada una se explica brevemente y de manera sencilla e intuitiva los algoritmos del material original. En algunos casos, sólo aquellos algoritmos que se consideran más relevantes serán mencionados.

El texto de la Sección 2.1 se menciona en este capítulo porque es de los primeros trabajos realizados en el área de protocolos de red distribuidos e introduce las ideas básicas de cómo resolver los problemas comunes en estos sistemas. Por otra parte, en la Sección 2.2 se muestran nuevas técnicas en algoritmos para la solución de algunos problemas comunes en sistemas distribuidos. En la Sección 2.3 se trata el tema de eficiencia energética en la difusión de mensajes en una red inalámbrica de sensores equipados con una antena direccional y una omnidireccional. Por último, el texto de la Sección 2.4 proporciona algoritmos para el descubrimiento de vecinos en redes inalámbricas de sensores con antenas direccionales.

2.1. Protocolos de Red Distribuidos

En el artículo, Segall [8] considera la situación en donde un número de unidades físicas de cómputo distintas trabajan sobre un problema en común y su operación es coordinada mediante canales que conectan algunas de estas unidades. El programa que reside en cada nodo se conoce como *algoritmo de nodo* y la unión de todos los algoritmos que proveen la solución al problema se denomina *protocolo distribuido*.

El autor considera protocolos de red distribuidos que tienen como principal aplicación la comunicación de datos entre dispositivos dispersos geográficamente que deben transmitir información de manera coordinada a algún otro. Con esta aplicación en mente, los protocolos han sido propuestos para los problemas de ruteo y propagación de información (*broadcasting information*), pruebas de conectividad y la búsqueda de rutas más cortas.

En una red, los nodos y enlaces pueden fallar o ser agregados, de manera que, los protocolos deben ser capaces de trabajar bajo cualquier cambio arbitrario en la red. El autor primero considera aquellos protocolos para una red con topología fija, que después se extienden para considerar posibles cambios.

Segall define a la red como el par ordenado (V, E) , donde V es el conjunto de nodos y E el conjunto de enlaces o aristas entre los nodos. Cada arista es bidireccional y se asegura que cada mensaje enviado por un nodo i llegará correctamente en un tiempo finito distinto de cero a su destino j ; además de que todos los mensajes se reciben en el mismo orden de envío. Los mensajes que recibe un nodo i se etiquetan con un identificador del enlace del cual llegaron y se transfieren a una cola común. Adicionalmente, cada nodo posee un identificador y sabe exactamente los identificadores de cada nodo en la red, así como el número de vecinos que tiene. Sin embargo, no conoce el identificador correspondiente a cada vecino, ni que nodos pertenecen a la red en un momento determinado de la ejecución. Por otra parte, varios nodos pueden comenzar la ejecución de cualquier protocolo de manera asíncrona.

2.1.1. Propagación de información

En esta sección se explica la implementación del autor de dos algoritmos para la propagación de información, denominados: *PI (Propagation of Information)* y *PIF (Propagation of Information with Feedback)*.

El primer algoritmo, PI, lleva a cabo una inundación del mensaje en la red, es decir, cada vez que un vértice recibe el mensaje lo envía a todos sus vecinos, que a su vez, cada uno de ellos lo envía a todos sus vecinos y así sucesivamente. Cada vértice procesará el primer mensaje que reciba de sus vecinos ignorando aquellos que lleguen después.

El segundo protocolo, PIF, agrega al protocolo anterior un proceso de retroalimentación (*feedback*). La idea de inundación del protocolo PI se sigue manteniendo, pero ahora el protocolo se ejecuta en dos fases u *olas*: en la primera se transmite el mensaje del nodo que inicia la ejecución hacia los otros nodos restantes de la red y en la segunda, se transmite de los nodos restantes hacia el nodo que inició la ejecución. Con este proceso se confirma al nodo inicial que la información se ha transmitido por la red. En este caso, se supone que sólo un nodo comienza la ejecución del protocolo.

2.1.2. Pruebas de conectividad

En esta sección se muestra la solución del autor para llevar a cabo *pruebas de conectividad* en una red, que significa que mediante un algoritmo cada nodo puede saber qué nodos son *alcanzables* en la red para él, es decir, qué nodos pueden recibir información desde él en un determinado momento.

Se proponen algunos protocolos para estas pruebas. Los principales son: *CT1 (Connectivity Test 1)* y *CT2 (Connectivity Test 2)* basados en los algoritmos de propagación de información PI y PIF. El algoritmo *CT3 (Connectivity Test 3)* está basado en CT1, pero incluye la *propiedad de término*, que permite a los nodos saber que el protocolo ha sido completado. Los otros algoritmos, *CT4 (Connectivity Test 4)* y *CT5 (Connectivity Test 5)*, están basados en CT3 y son versiones más eficientes.

El primer protocolo, CT1, se basa en el uso repetido del protocolo PI. La primera vez se usa para informar a todos los nodos que el protocolo está en

progreso y después para que un nodo propague su identidad. Uno o varios nodos pueden comenzar la ejecución del protocolo si reciben un mensaje de COMIENZO, o bien, si reciben un mensaje de control desde alguno de sus vecinos. En cualquiera de los dos casos, lo primero que hace un nodo es enviar un mensaje de control con su identificador y así, permitir la propagación del mismo. En el caso de que el nodo reciba un mensaje de control de algún vecino, guarda el identificador del nodo que lo envió y así sabe que está conectado a él. Después, el nodo sigue transmitiendo este mensaje para que los demás nodos guarden también esta información. Lo anterior se hace con cada mensaje de control que se recibe de cualquier vecino.

Con este protocolo no hay manera de que un nodo sepa con seguridad cuáles nodos están desconectados de él. En otras palabras, no hay manera de que el nodo sepa cuándo el algoritmo se completó, al menos que todos los nodos estén conectados. Para solucionar este problema se proporcionan otros protocolos que permiten a los nodos tener este conocimiento.

El segundo protocolo, CT2, se basa en el protocolo PIF. El protocolo comienza de la misma manera que CT1 y usa PIF repetidamente con el objetivo de que cada nodo envíe su identificador a la red. Cada nodo i contiene variables para cada uno de los nodos en la red, que tendrán un valor de 1 para aquellos nodos conectados a él y 0 para los desconectados. Se asegura que mediante este protocolo, el nodo i al recibir de regreso el mensaje que contiene su propio identificador desde todos sus vecinos, podrá distinguir entre el conjunto de nodos al cual está conectado y el conjunto de nodos al cual no tiene una conexión.

En los protocolos siguientes se utilizan ideas un poco distintas para llevar a cabo la propiedad de término. El protocolo CT3 sirve de base para los dos protocolos más eficientes, CT4 y CT5. CT3 se basa en el protocolo CT1, con la diferencia de que en lugar de sólo enviar mensajes con el identificador del nodo se incluye el identificador de todos sus vecinos, para que al recibir un mensaje un nodo j del nodo i , el nodo j deba esperar recibir el mensaje de todos los nodos vecinos de i . De esta manera, la señal de término ocurrirá cuando el nodo j haya recibido un mensaje de todos los vecinos de cada uno de sus vecinos. Esta versión tiene más requerimientos en la comunicación, así que las versiones CT4 y CT5 mejoran su desempeño incluyendo otro tipo de variables.

Los algoritmos CT4 y CT5 salen del alcance de este texto, al ser cada vez más específicos e incluir más variables que pueden complicar su descripción.

2.1.3. Trayectorias más cortas

En esta sección se presenta el algoritmo propuesto por Segall [8] y que obtiene la *trayectoria más corta* desde un nodo en la red a cualquier otro, es decir, la trayectoria con el mínimo número de enlaces. Este protocolo es denominado *MH (Minimum-Hop Path)*.

Cuando comienza a ejecutarse el protocolo, un nodo i envía su identificador a todos sus vecinos, con la finalidad de recibir después el identificador de todos ellos y registrarlos a una distancia 1 de él. Esta información la envía de nuevo a sus vecinos y espera recibir una lista de cada uno de ellos, que contiene los identificadores de los nodos que tienen a distancia 1. Así, el nodo i puede registrar estos nodos a distancia 2 de él. Este proceso se repite hasta que la unión del conjunto de los identificadores de los nodos conocidos por i con el nuevo conjunto recibido no contiene nodos desconocidos. Entonces, el nodo i ha completado en ese momento el algoritmo y cualquier otro mensaje que se reciba será ignorado.

2.1.4. Cambios topológicos

El autor explica cómo se ataca el cambio topológico de una red cuando un protocolo está en ejecución. Este tipo de cambios pueden darse ya sea porque se presenta alguna falla o una recuperación de un enlace o de un nodo. Para esto, el artículo ofrece modificaciones a los protocolos anteriores para que en la red puedan existir cambios en la topología y sigan funcionando de manera correcta.

La idea general es que cuando un cambio topológico sea detectado en algún nodo, un nuevo ciclo del protocolo se inicie para informar a la red de la nueva situación. Los ciclos del protocolo están etiquetados mediante números enteros crecientes y cada nodo recuerda el ciclo mayor. Cuando un nodo desea iniciar un nuevo ciclo como resultado de un cambio topológico en un enlace adyacente, reinicia los valores de sus variables e incrementa el número de su ciclo. Este número se envía con todos los mensajes, de modo que, cuando un

nodo reciba un mensaje con un número de ciclo más pequeño, lo ignorará. Además reinicia sus variables, actualiza su número de ciclo y actúa como si comenzara a ejecutarse de nuevo el protocolo.

Con base en estas modificaciones, se tiene lo siguiente:

- El enlace (i, j) falla o se recupera al mismo tiempo que el enlace (j, i) falla o se recupera;
- Cada mensaje enviado por un nodo i y por el enlace (i, j) , llega correctamente a j en tiempo finito distinto de cero o el enlace falla en tiempo finito;
- Ambos nodos finales son notificados en tiempo finito cuando un enlace falle o se recupere, pero no necesariamente al mismo tiempo;
- La falla o recuperación de un nodo es considerado falla o recuperación, según sea el caso, de todos los enlaces adyacentes al nodo.

2.2. Algoritmos de inversión de enlaces

Welch et al. [14] diseñan algoritmos de inversión de enlaces, que ha sido una técnica muy utilizada en diversos algoritmos distribuidos para una variedad de problemas en redes como: ruteo de información, elección de un líder, exclusión mutua, asignación de recursos y encolamiento distribuido.

Los autores modelan al sistema distribuido como una gráfica con los vértices representando los nodos y las aristas, alguna característica del sistema. La idea es que en cada algoritmo se asigne una orientación virtual a las aristas, lo que da como resultado una versión dirigida de la gráfica original. Estas orientaciones virtuales pueden cambiar para poder cumplir algún objetivo específico del algoritmo.

Aquí se dará una idea en general de cómo se utilizan estos algoritmos para resolver el problema de ruteo de información y el de exclusión mutua.

2.2.1. Ruteo en un sistema distribuido

En esta subsección se explica el funcionamiento de los algoritmos de ruteo de información que proponen los autores, quiénes consideran un único nodo destino de la información y al que los otros nodos en la gráfica necesitan mandar mensajes. El objetivo es asignar orientaciones a las aristas de la gráfica que permitan reenviar mensajes al nodo destino.

El primer algoritmo lo denominan algoritmo de Inversión Total (*Full Reversal*) y el segundo, algoritmo de Inversión Parcial (*Partial Reversal*). Estos algoritmos suponen que no existen cambios topológicos de la gráfica durante la ejecución. La idea principal en estos algoritmos es que, cualquier vértice distinto del vértice destino que sea un *sumidero*¹ (todas las aristas incidentes en él apuntan a él) invierte algunas o todas las orientaciones de las aristas incidentes en él, de manera que, los algoritmos van a diferir en cómo se escogen estas aristas.

El problema a resolver se convierte en cambiar la orientación de cierto subconjunto de aristas de la gráfica, de manera que exista un camino de cada vértice al vértice destino D . La gráfica que se obtiene como resultado la denominan *gráfica D -orientada*.

Los autores definen un algoritmo base denominado *Algoritmo de Inversión de Enlaces Genérico (GLR)*. Este algoritmo, en cada iteración, escoge un subconjunto de vértices sumideros, excepto a D en caso de serlo, e invierte algunas de las aristas incidentes en ellos. Este algoritmo no cumple en todos los casos con la propiedad de término y esto dependerá de como se escogen los vértices sumideros y sus aristas a invertir. Un ejemplo de cuando el algoritmo no termina sería considerar tres vértices: D , x y y , donde las orientaciones de las aristas se ven de la siguiente manera:

$$D \rightarrow x \leftarrow y$$

Supongamos que x invierte su arista derecha y después y invierte su arista izquierda, de nuevo x invierte su arista derecha y así sucesivamente.

Una primera propuesta de los autores para asegurar que el algoritmo termina es el algoritmo de inversión total, que propone que para el subconjunto

¹Se usa el término sumidero como traducción literal de *sink*

de vértices sumideros elegidos en cada iteración se inviertan todas las aristas incidentes en ellos. Este algoritmo cumple con la propiedad de término para cualquier caso y se demuestra que existe un tiempo t en el que un vértice que era sumidero jamás volverá a serlo. Como es de suponer, durante la ejecución algunas aristas pueden invertirse repetida e innecesariamente, por lo que el algoritmo de inversión parcial intenta reducir el número de inversiones. La inversión de las aristas en este algoritmo se hace de la siguiente forma: para cada sumidero v del subconjunto escogido en una iteración, una arista (u, v) es invertida por v si y sólo si, la arista no fue invertida por u la última vez que fue escogido dentro del subconjunto de sumideros. En el algoritmo cada vértice mantiene una lista indicando que aristas invirtió la última vez. Al igual que el algoritmo anterior, se demuestra que existe un tiempo t a partir del cual un vértice que era sumidero no volverá a serlo.

Se proponen otros algoritmos basados en los dos algoritmos anteriores y que además llevan a cabo un etiquetado de vértices o aristas para indicar la dirección en la que se transmite información.

2.2.2. Exclusión mutua en un sistema distribuido

Los autores consideran algoritmos basados en la inversión de enlaces que resuelven el problema de exclusión mutua. El algoritmo que se propone es denominado *LRME (Link Reversal Mutual Exclusion)*.

El problema de exclusión mutua se define de la siguiente manera: cada nodo ejecuta un proceso de aplicación que tiene una sección especial de código llamada *sección crítica*. Cuando un proceso de aplicación desea entrar a esta sección debe esperar a ser notificado de que ya puede entrar, ya que dos procesos de aplicación no pueden ejecutarla al mismo tiempo. Mientras un proceso de aplicación está esperando para poder entrar a esta sección, se dice que se encuentra en su *sección de espera*. Cuando un proceso no necesita entrar en su sección crítica se dice que está en su *sección restante*.

Cada nodo ejecuta un proceso de exclusión mutua. Este proceso y el de aplicación dentro del mismo nodo cooperan de manera que el proceso de aplicación haga un ciclo entre su sección restante, su sección de espera, su sección crítica y regrese a su sección restante. El conjunto de procesos de exclusión mutua se comunican entre ellos por medio de paso de mensajes sobre

la red para sincronizar cuándo los procesos de aplicación pueden entrar a su sección crítica, por lo que los procesos de exclusión mutua deben asegurar que en cada punto de la ejecución a lo más un proceso de aplicación está dentro de su sección crítica y que cada proceso que requiera entrar a esta sección, eventualmente entrará.

La exclusión se asegura al mantener un indicador en el sistema y que circula en mensajes entre los nodos de la gráfica. Un nodo que requiere entrar a su sección crítica puede hacerlo sólo si tiene este indicador.

El algoritmo propuesto por Welch et al. [14] funciona de la siguiente manera: cada nodo i tiene una variable local llamada $status_i$, que toma uno de los siguientes valores: restante, espera y crítica, dependiendo de la sección en la que el proceso de aplicación se encuentre. Además, i tiene otra variable local, denominada N_i , que contiene los identificadores de todos sus vecinos en la gráfica.

Para manipular el indicador, cada nodo i tiene una variable local llamada $hasToken_i$, que tiene el valor de verdadero si y sólo si i tiene el indicador. Por otro lado, para manejar las orientaciones de los enlaces, i tiene otras dos variables locales llamadas IC_i y OG_i , que contienen los identificadores de los nodos vecinos que son *entrantes* y *salientes* respectivamente. Un nodo vecino j es nodo vecino entrante de i cuando la orientación del enlace que los conecta apunta hacia i y saliente cuando esta orientación apunta a j . Se supone que las variables IC_i y OG_i se inician de manera que la variable $hasToken$ de exactamente uno de los nodos en la gráfica tiene valor verdadero.

Cuando un nodo i recibe un mensaje que contiene el indicador cambia el valor de su variable $hasToken_i$ a verdadero e invierte las orientaciones de todos sus enlaces salientes, convirtiéndose así, en un nodo sumidero. El mecanismo de inversión de enlaces se maneja por medio de mensajes y se realiza de la siguiente forma: el nodo i envía un mensaje llamado *MAKEOG* a cada vecino saliente j , indicándole que mueva a i de su variable IC_j hacia su variable OG_j . Cuando j lo hace manda un mensaje de confirmación a i , denominado *AckOG*. Una vez que i recibió el mensaje de confirmación de todos sus vecinos salientes pone a todos sus vecinos en su variable IC_i y vacía OG_i .

Por otra parte, para asegurar un acceso justo a la sección crítica, cada nodo i mantiene una variable local llamada RQ_i , la cual corresponde a una cola que guarda el indicador de él o de los nodos vecinos que han hecho una petición del indicador. Esta cola se inicia vacía y cuando i desea entrar a su sección crítica guarda su identificador en ella. De manera similar, cuando un nodo i recibe un mensaje de petición del indicador desde un nodo entrante j , guarda el identificador de j en RQ_i . En ambos casos, si el identificador que se agrega a la cola de i es el único, entonces i manda un mensaje de petición del indicador a cualquiera de sus vecinos salientes. En caso de no ser el único, no manda mensaje y confía en que el envío previo, cuando RQ_i era vacía, atraerá el indicador hacia él.

Un nodo i que tiene actualmente el indicador ejecuta la función que lo maneja, denominada *handleToken()*, en 3 situaciones distintas:

1. si i recibe un mensaje de petición del indicador mientras no está en su sección crítica;
2. si i entra en su sección de espera;
3. si i deja su sección crítica y su variable RQ_i no es vacía.

En cada caso, al ejecutar la función el nodo i saca un elemento de RQ_i y si corresponde a su identificador entra a su sección crítica, si no, envía el indicador en un mensaje al nodo cuyo identificador corresponde al del elemento y entonces, si la cola no quedó vacía, envía un mensaje para pedir de nuevo el indicador.

En el texto los autores demuestran que el algoritmo LRME realmente resuelve el problema de exclusión mutua.

2.3. Eficiencia energética en la difusión de mensajes en una red inalámbrica de sensores con antenas direccionales

Li et al. [5] proponen un algoritmo para la difusión de mensajes en este tipo de redes que realiza un consumo energético menor que algoritmos diseñados anteriormente. La idea del algoritmo es que al surgir una petición para la difusión de un mensaje desde un nodo s en la red, se encuentre el árbol con raíz en s , en donde los nodos seleccionan su área de transmisión de tal modo que el consumo de energía sea minimizado.

Cada nodo está equipado con una antena direccional para el envío de mensajes y una omnidireccional para la recepción. Se supone que las señales de recepción no cuestan energía adicional, así que el consumo de energía total del árbol es la suma del consumo de energía de los nodos que envían señales, es decir, los nodos que no son hojas.

El algoritmo usa tres conjuntos de nodos diferentes. El primero es un conjunto de nodos cubiertos, llamado C , que contiene a los nodos que ya forman parte del árbol de difusión. El algoritmo consiste en encontrar el conjunto C que lleve a cabo un ahorro energético y cubra todos los nodos de la red. El segundo es N , el conjunto de los nodos candidatos a formar parte de C y que consiste en la unión de los vecinos de los nodos en C . En cada paso del algoritmo, un nodo en N será seleccionado para ser incluido en C . El tercer conjunto es U , de nodos no cubiertos.

El algoritmo forma el árbol de difusión desde s , de manera que, al comienzo el conjunto C es vacío y U tiene a todos los nodos excepto a s , que está dentro de N . A continuación, un nodo en N se selecciona y se incluye en C , sus nodos vecinos se borran de U y se agregan a N . Esta operación se repite hasta que U es vacío, lo que significa que todos los nodos en N han sido cubiertos y los nodos en C son los nodos en el árbol de difusión.

Para escoger el área de transmisión y seleccionar un nodo en N que será agregado a C se usan dos funciones, una que evalúa cada área de propagación de un nodo candidato y otra que da una evaluación total del mismo. Al final, el candidato con mejor evaluación será el próximo a agregar en C .

Mediante simulaciones los autores concluyen que el algoritmo brinda un mayor ahorro de energía comparado con algoritmos que ya han sido propuestos con anterioridad.

2.4. Descubrimiento de vecinos en una red de sensores con antenas direccionales

Kranakis et al. [2] consideran el problema de descubrimiento eficiente de vecinos en una red inalámbrica de sensores síncrona que emplea antenas direccionales para recibir o enviar mensajes. En el modelo, el área de transmisión de la antena direccional corresponde a un sector en un disco circular y los nodos sensores pueden rotar su antena en sentido contrario a las manecillas del reloj como sea requerido para descubrir a los vecinos. Además, dos nodos pueden comunicarse cuando sus antenas correspondientes están orientadas de manera que cada nodo sensor esté dentro del rango de transmisión del otro.

Los autores proponen una clase de algoritmos para la rotación de antenas que sirven de base para los algoritmos del descubrimiento de vecinos y estudian su eficiencia energética mediante simulaciones.

2.4.1. Algoritmos deterministas

En este tipo de algoritmos los autores explotan el conocimiento de una mínima coloración de los vértices de la red. Es decir, una coloración de la red empleando el mínimo número de colores.

El primer algoritmo determinista, denominado *ARA* (*Antenna Rotation Algorithm*), define la rotación de la antena de los nodos en la red de la siguiente manera: cada nodo u tiene una variable d_u , la cual indica el tiempo que su antena debe mantener la transmisión sobre cualquiera de sus sectores. Durante este tiempo, el nodo u intentará enviar o recibir mensajes a o de otros nodos que pudieran alinear su antena de manera adecuada para poder comunicarse. Después de transcurrido este tiempo, el nodo u cubre un nuevo sector girando su antena cierto número de grados en sentido contrario a las

manecillas del reloj y repite el proceso anteriormente descrito. La antena de cada nodo da un número indefinido de vueltas, cubriendo en cada una de ellas a cada uno de sus sectores. El algoritmo ARA inicia al mismo tiempo esta rotación para cada antena en la red y dependiendo del tiempo d_u de cada nodo, puede ser que dos nodos vecinos puedan encontrarse en algún momento a causa de esta rotación y entonces se comuniquen mediante la recepción o el envío de mensajes.

Los algoritmos para el descubrimiento de vecinos se basan en el algoritmo ARA y la idea principal de ellos es escoger de manera adecuada los tiempos d_u y d_v para cada par de nodos vecinos u y v , que aseguren que en determinado momento ambos nodos sensores podrán comunicarse ejecutando el algoritmo ARA. La selección de los tiempos también depende del número de sectores que tenga un nodo sensor. Por ejemplo, los autores suponen en uno de sus teoremas que todos los nodos tienen k sectores y proponen escoger a cada d_u de acuerdo a la coloración conocida de la red. Suponiendo que la coloración mínima se hace con c colores, el tiempo de transmisión para cada nodo u se define como $d_u = k^{c_u}$, donde c_u corresponde al color asignado al nodo u . Con esto se demuestra que al ejecutar el algoritmo ARA, cada nodo sensor descubrirá a todos sus vecinos en a lo más k^{c-1} unidades de tiempo.

Cuando el número de sectores en cada nodo es distinto, los autores proponen otra manera de escoger los tiempos para asegurar la comunicación entre dos nodos vecinos. Esta elección consiste en lo siguiente: cada nodo u tiene k_u sectores y un tiempo d_u de transmisión, entonces al escoger tanto a d_u como un número primo y $d_u > k_u$, así como a d_u y d_v primos relativos, para cualquier tiempo d_v de un nodo vecino v de u , los autores demuestran que al ejecutar el algoritmo ARA cada nodo sensor en la red descubrirá a todos sus vecinos en a lo más $O((\max_u k_u)^2 \cdot (\max_u d_u)^2)$ unidades de tiempo.

2.4.2. Algoritmos de descubrimiento de vecinos aleatorios

La principal ventaja de estos algoritmos es que simplemente se necesita una cota superior n del número de nodos en la red. Los dos algoritmos principales que proponen los autores son: el algoritmo determinista con selección aleatoria del tiempo de transmisión d_u para el nodo u , *RARA*, y el algoritmo con selección aleatoria del mecanismo de rotación, *RSRMA*.

En el algoritmo RARA, cada nodo sensor u tiene k_u sectores, selecciona un número primo aleatorio d_u entre k_u y n como tiempo de transmisión y con estos valores ejecuta el algoritmo determinista ARA. Mediante este algoritmo y el conocimiento de n , se asegura que cada sensor en la red descubrirá a sus vecinos con alta probabilidad.

Para el caso cuando todos los nodos tienen k sectores, los autores proponen el algoritmo RSRMA. Este algoritmo consiste en que un nodo sensor escoge de manera aleatoria un mecanismo de rotación de dos disponibles. En el primer mecanismo, la antena gira k veces con un tiempo de transmisión de una unidad por sector, mientras que en el segundo, la antena gira sólo una vez pero con un tiempo de transmisión de k unidades de tiempo por sector. Con este algoritmo se asegura que si cada nodo sensor en la red lo ejecuta durante $O(\log n)$ veces, descubrirá a cada uno de sus vecinos con alta probabilidad.

Mediante simulaciones los autores concluyen que los algoritmos aleatorios llevan a cabo una mayor eficiencia energética y en general, que los algoritmos de descubrimiento de vecinos propuestos usando antenas direccionales reducen el consumo de energía al compararlos con el uso de antenas omnidireccionales.

Capítulo 3

Difusión de mensajes

En este capítulo se analiza el problema de difusión de mensajes en una red inalámbrica de sensores con antenas direccionales, la cual está compuesta de nodos sensores capaces de proporcionar comunicación inalámbrica, ya que poseen antenas que radian energía hacia un área que depende del tipo de antenas que se utiliza. Además, estos nodos están compuestos de un procesador, memoria, batería y sensores, tal como se mencionó en la Sección 1.1. La red se representa como se definió en la Subsección 1.3.2, es decir, mediante una gráfica en donde los vértices simbolizan a los nodos y las aristas a los enlaces de comunicación .

También se plantea el problema a resolver y se proporcionan dos algoritmos para llevar a cabo la difusión. El primer algoritmo, denominado *ADM*, lleva a cabo la difusión de un mensaje y el segundo, denominado *AGDM*, crea un árbol de difusión de la gráfica que se utiliza en el primer algoritmo. Se presenta primero el algoritmo de difusión para comprender mejor la construcción del árbol en el algoritmo *AGDM*.

3.1. Planteamiento del problema

Con el modelo de red proporcionado en la Sección 1.3, el problema de difusión de mensajes puede transcribirse de la siguiente manera: dada una gráfica $G = (V, E)$ que representa a una red inalámbrica de sensores con antenas direccionales y un mensaje M que necesita difundirse, se tiene que al surgir la petición de difusión de M desde un nodo $u \in V$, se debe enviar a M a cada uno de los nodos en V distintos de u .

3.2. Algoritmos para la difusión de mensajes

En esta sección se presentan dos algoritmos que en conjunto llevan a cabo la difusión de mensajes. En un principio se define el algoritmo de difusión de mensajes, que supone que se ha generado un árbol de difusión de la gráfica. Después se define el algoritmo generador de dicho árbol de difusión.

El algoritmo ADM se basa en dos algoritmos de rotación de antenas definidos por Kranakis et al. [2]. En ADM se hicieron unas pequeñas modificaciones a los algoritmos para que los nodos no transmitieran energía por sectores innecesarios.

Li et al. [5] definen un algoritmo similar al algoritmo AGDM y en el cual utilizan un modelo de comunicación (D, O) , es decir, el nodo emisor posee una antena direccional y el receptor una omnidireccional. En el algoritmo que presentan los autores se crea un árbol de difusión que cuida el gasto de energía mediante funciones de evaluación de los nodos, las cuales definen el orden que llevan los nodos en el árbol de acuerdo a este gasto. AGDM fue diseñado con la misma idea de construir un árbol de difusión tomando en cuenta que el modelo de comunicación es (D, D) y cambiando la evaluación de los nodos.

3.2.1. Algoritmo de difusión de mensajes

El algoritmo ADM, presentado en la Figura 3.1, utiliza un árbol de difusión $A = (V, E')$ de la gráfica original para realizar la difusión de un mensaje M . Esta difusión comienza desde el nodo raíz, que es en realidad el único nodo que posee el mensaje M al comenzar la ejecución.

El algoritmo utiliza dos tipos de rotación de antena y el tipo que ocupa un nodo en determinado momento depende de si ya recibió o no el mensaje M .

El primer tipo de rotación se ejecuta cuando un nodo u aún no recibe el mensaje M . La antena de u transmite energía por cada uno de sus sectores durante una unidad de tiempo (líneas 16 a 18). En cada sector, u intenta comunicarse con el nodo vecino v que debe enviarle a M (línea 17), así que envía un mensaje con su identificador (línea 17) para que si v lo recibe, pueda identificar a u como uno de sus vecinos en el árbol y entonces le envíe

una copia de M . En este caso se supone que v ya recibió el mensaje M con anterioridad y si no es así, simplemente ignorará el mensaje de u , al igual que en el caso donde v no identifique a u como su vecino.

Por ejemplo, en la Figura 3.2 se muestra una gráfica que representa a una red y su árbol de difusión correspondiente. En este árbol, el nodo v_1 espera recibir a M del nodo v_0 y cuando logre comunicarse con él, v_0 lo identificará como su vecino en el árbol y le enviará una copia de M .

ADM($A = (V, E')$)

```

1: para cada nodo  $u \in V$  hacer
2:   Comenzar transmisión sobre un determinado sector  $i$ ;
3:   mientras  $u$  tenga un vecino en  $A$  que no haya recibido el mensaje  $M$ 
   hacer
4:     si  $u$  ha recibido el mensaje entonces
5:       para  $j := 0$  hasta  $d_{u_i} - 1$  hacer
6:         Recibir identificador de vecino(s); //Si hay vecinos disponibles
7:         Identificar vecino(s);
8:         si Identifica vecino(s) en el árbol entonces
9:           Enviar copias del mensaje  $M$  a vecino(s); //Si hay vecinos
           disponibles
10:        si no
11:          Rechazar mensaje de vecino(s);
12:        fin si
13:      fin para
14:      Rotar antena hacia el siguiente sector  $i$  en el árbol;
15:    si no
16:      Enviar identificador a vecino(s);
17:      Recibir copia del mensaje  $M$  de vecino; //Si hay vecino disponible
18:      Rotar antena  $\phi_u$  grados en sentido contrario a las manecillas del
      reloj para cubrir un nuevo sector  $i$ ;
19:    fin si
20:  fin mientras
21: fin para

```

Figura 3.1: Algoritmo para la Difusión de Mensajes (ADM)

Gracias a que la estructura con la que el algoritmo funciona es un árbol de difusión, cada nodo en la red sabe perfectamente a qué vecinos tiene

que enviar a M y por cuáles sectores. De modo que, la antena de los nodos no necesita transmitir por cada uno de sus sectores. Entonces, el otro tipo de rotación se define de la siguiente manera: después de que u recibe a M , su antena intentará transmitirlo (línea 9) solamente por cada sector i con vecinos en el árbol (línea 14) y mantendrá la transmisión durante las d_{u_i} unidades de tiempo (líneas 5 a 13), con la finalidad de que los nodos vecinos tengan suficiente tiempo para dirigir su antena hacia u y puedan comunicarse.

Tomando en cuenta de nuevo la Figura 3.2 y el último tipo de rotación mencionado, la antena del nodo v_0 sólo transmitirá por los sectores cero y uno, pues tiene en el árbol a v_1 y a v_2 como vecinos en esos sectores. Sin embargo, no transmitirá por el sector dos, aunque en la gráfica original tiene al nodo v_3 como vecino por ese sector.

Para el propósito de este texto es suficiente suponer que el proceso de intercambio de mensajes se lleva a cabo en una unidad de tiempo, en la cual, el nodo sensor u envía su identidad al nodo v y v le contesta con una copia de M , en caso de tener el mensaje.

La difusión correcta del mensaje se determina por los tiempos d_{u_i} asignados a cada sector i de un nodo u en la red, ya que de cada tiempo depende si u puede encontrarse con sus vecinos en el árbol. Entonces, la elección del valor de estas variables es importante para asegurar la comunicación de dos nodos vecinos en A . El algoritmo AGDM, definido en la siguiente sección, se encarga también de escoger el valor de las variables d_{u_i} para cada nodo u que asegure la comunicación con sus vecinos.

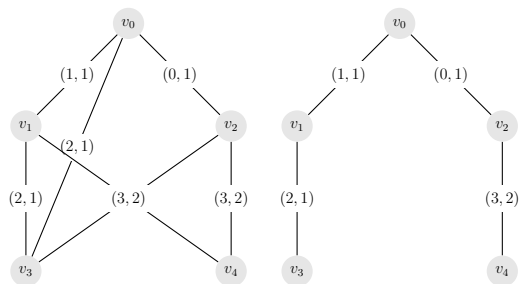


Figura 3.2: Gráfica G y su árbol de difusión

3.2.2. Algoritmo AGDM

El objetivo del algoritmo es definir las trayectorias que el mensaje debe seguir para llegar a cada uno de los nodos.

El algoritmo AGDM, descrito en la Figura 3.3, funciona con una gráfica no dirigida y conexa $G = (V, E)$ que representa la red de sensores y con un nodo inicial $v \in V$ al que llega la petición de difusión de un mensaje. El algoritmo mantiene dos conjuntos de nodos: el conjunto A que contiene los nodos incluidos en el árbol y el conjunto R que contiene a los nodos restantes. En un comienzo, el conjunto A contiene solamente a v (línea 1), que será el nodo raíz del árbol.

AGDM determina para cada nodo u incluido en el árbol, el sector s_u que tiene el mayor número de vecinos no incluidos en el árbol (línea 6). A ese conjunto de nodos se le denomina $N_{s_u}(u)$. Además, en una estructura denominada C se conservan solamente los conjuntos $N_{s_u}(u)$ con la mayor cardinalidad y se almacena el valor de esta cardinalidad en una variable llamada *cardinalidadConjuntos*.

Cuando se determine un conjunto $N_{s_u}(u)$ se verificará si su cardinalidad es mayor a la cardinalidad de los conjuntos en C (línea 7). En ese caso, se vacía C , pues se ha descubierto un conjunto con una cardinalidad mayor a los ya existentes en la estructura, se agrega el conjunto $N_{s_u}(u)$ a C y se actualiza el valor de la variable *cardinalidadConjuntos* de C (líneas 8 a 10). En el caso en que la cardinalidad sea igual a la de los conjuntos de C (línea 12), solamente se agrega el conjunto $N_{s_u}(u)$ a C (línea 13).

Después de ejecutarse el proceso anterior para cada uno de los nodos existentes en A , los conjuntos que quedan en C tienen la misma y la mayor cardinalidad. Luego, la función *eval* calcula el consumo de energía para comunicarse con los nodos de cada conjunto y escoge aquél que requiera menos (línea 17), denominado en el algoritmo como *conjuntoCandidato_v*, pues indica que v es el nodo desde el cual se podrá tener comunicación con los nodos de este conjunto.

```

ADM(Gráfica  $G, v$ )
1: Conjunto  $A := \{v\}$ ;
2: Conjunto  $R := G.nodos - \{v\}$ ;
3:  $C := \emptyset$ ;
4: mientras  $R$  no es vacío hacer
5:   para cada  $u \in A$  con vecinos en  $R$  hacer
6:     Determinar  $S_u$  para el nodo  $u$ ; //Obtiene el sector de la antena de
        $u$  con más nodos
7:     si  $C$  es vacío  $\vee |N_{s_u}(u)| > C.cardinalidadConjuntos$  entonces
8:        $C := \emptyset$ ;
9:        $C.agregaConjuntoCandidato(N_{s_u}(u))$ ;
10:       $C.cardinalidadConjuntos = |N_{s_u}(u)|$ ;
11:     si no
12:       si  $|N_{s_u}(u)| = C.cardinalidadConjuntos$  entonces
13:          $C.agregaConjuntoCandidato(N_{s_u}(u))$ ;
14:       fin si
15:     fin si
16:   fin para
17:    $conjuntoCandidato_v := eval(C)$ ; //Conjunto de nodos que consume
       menos energía
18:    $d_{vs_v} = max\{k_{v_i}\}$ ;  $v_i \in conjuntoCandidato_v$ 
19:    $A := A \cup conjuntoCandidato_v$ ;
20:    $R := R - conjuntoCandidato_v$ ;
21: fin mientras

```

Figura 3.3: Algoritmo Generador del Árbol de Difusión de Mensajes (AGDM)

El siguiente paso del algoritmo es determinar un valor para la variable d_{vs_v} (línea 18). Como se definió en la Sección 1.3, esta variable indica el tiempo que la antena del nodo v mantendrá la transmisión por su sector s_v . Asimismo, en esa sección se define que cada nodo u en la red posee una variable k_u que indica su número de sectores. El algoritmo se fija en el valor de esta variable para cada uno de los nodos en el conjunto $conjuntoCandidato_v$ y escoge la que tenga el máximo valor para asignarlo a la variable d_{vs_v} . La razón de establecer ese valor a la variable es que al ejecutar el algoritmo ADM, definido en la Subsección 3.2.1, se asegura que el nodo v espera el suficiente tiempo para que la antena de cada uno de sus nodos vecinos en el árbol transmita durante una unidad de tiempo por cada sector. De esta manera, v puede comunicarse con ellos y difundir el mensaje.

Para finalizar, se agregan los nodos del conjunto candidato al conjunto A para que pasen a formar parte del árbol (línea 19) y se borran del conjunto R (línea 20).

Todos los procesos anteriores se repiten hasta que el árbol contiene todos los nodos de la gráfica G (líneas 4 a 21).

Por otra parte, la energía necesaria para enviar un mensaje de un nodo u a un nodo v es la energía que se consume al transmitir durante d_{ui} unidades de tiempo por el sector i de u correspondiente para comunicarse con v . Esta cantidad de energía corresponde al área de transmisión de u multiplicada por d_{ui} . Por ejemplo, si suponemos que el nodo v_1 en la Figura 3.4a tiene un área de transmisión de 300 unidades y su variable $d_{v_1,3}$ tiene un valor de 5, el consumo total de energía para enviar un mensaje del nodo v_1 al nodo v_2 es de 1500 unidades de energía.

La evaluación de la función *eval* se basa en que si un nodo v puede comunicarse por alguno de sus sectores con dos o más vecinos, éste puede hacer copias del mensaje y enviar una a cada uno de ellos por el mismo consumo de energía. Supongamos que, como se muestra en la Figura 3.4b, el nodo v_1 del ejemplo anterior puede comunicarse además, con otro nodo v_3 por su sector tres. Con esta configuración, el consumo total de energía para enviar un mensaje de v_1 a ambos nodos podría ser de 1500 unidades y no de 3000, como podría pensarse, ya que el nodo v_1 podría enviar dos copias del mensaje, una a v_2 y otra a v_3 por 1500 unidades de energía.

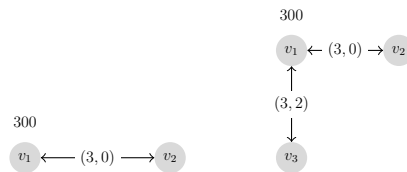


Figura 3.4: Consumo de 1500 unidades de energía para enviar un mensaje de v_1 a sus vecinos en el sector tres

Es importante garantizar que el algoritmo termina y construye el árbol adecuado para la difusión. El siguiente resultado demuestra que en efecto, eso sucede.

Teorema 1. *Sea $G = (V, E)$ una gráfica no dirigida y conexa que representa una red inalámbrica de sensores con antenas direccionales, donde cada nodo en V conoce a todos sus vecinos y sabe por cuál sector puede comunicarse con cada uno de ellos. Supongamos que surge una petición de difusión de un mensaje M desde un nodo $v_i \in V$ a todos los nodos de G . Entonces, al ejecutar el algoritmo AGDM con G y v_i como parámetros se formará un árbol de difusión con raíz en v_i , que contiene a todos los nodos de G y que proporciona las trayectorias, a partir de v_i , necesarias para transmitir a M a cada uno de los nodos en V .*

Demostración. Se demostrará que al inicio de cada iteración del ciclo de las líneas 4 a 21, el conjunto A contiene al menos un nodo nuevo que fue borrado de R , que en la gráfica formada no se forma ningún ciclo y que se trata de una gráfica conexa.

Esto puede observarse antes de la primera iteración, pues A contiene a v_i , el cual no se incluye en R . Asimismo, hasta el momento no se forma ningún ciclo en el árbol y éste es una gráfica conexa.

Mientras $R \neq \emptyset$, el cuerpo del ciclo funciona moviendo un subconjunto de nodos de R al conjunto A . Estos nodos son vecinos de algún nodo v_j que se encuentra en A desde alguna iteración anterior, por lo que el conjunto se denomina *conjuntoCandidato* $_{v_j}$. Este conjunto no es vacío, ya que G es una gráfica conexa, por lo que debe existir al menos un nodo en A que tenga un vecino que se encuentre en R , pues el ciclo no ha terminado. Entonces, antes de cada iteración en la cual $R \neq \emptyset$, se cumple que hay al menos un nuevo nodo en A , que ya no está en R . Además, cuando un nodo r es agregado a A como vecino de un nodo u , significa que solamente a través del nodo u podrá recibir el mensaje. Así que puede asegurarse que en las trayectorias hasta el momento formadas no existen ciclos, ya que cada nodo se agrega sólo una vez a A . También cuando se agrega a r se incluye un enlace de E que va desde u hacia este nuevo nodo, de manera que hasta el momento todos los nodos están conectados.

Finalmente, examinemos que sucede cuando el ciclo termina. La condición

para que termine es que $R = \emptyset$. Esto se logra ya que en cada iteración se borra al menos un nodo de R , por lo que en alguna iteración $k < |V| - 1$, R estará vacío y entonces todos los nodos de G estarán en A , lo que indica que hay una trayectoria para cada uno de los nodos desde v_i y el mensaje podrá seguir estas trayectorias y ser entregado a todos ellos en un tiempo finito. Al ser $R = \emptyset$, todos los nodos de G han sido agregados a la nueva gráfica, donde todos están conectados y no existen ciclos, por la razón anteriormente descrita. Por lo tanto, se ha formado un árbol con el mismo número de nodos de G que define las trayectorias para mandar a M a cada uno de los nodos en un tiempo finito. □

Después de mostrar que el algoritmo *AGDM* forma un árbol de difusión de una gráfica G , se procede a demostrar que el algoritmo *ADM* lleva a cabo de manera correcta la difusión de un mensaje. Para ello, bastará con probar que un nodo que recibe el mensaje puede difundirlo a sus nodos vecinos en el árbol. El siguiente resultado tiene como objetivo mostrar que con la asignación de los tiempos de transmisión en el algoritmo *AGDM* se logra que un nodo sea capaz de comunicarse con sus vecinos correspondientes y por lo tanto, de difundir el mensaje.

Teorema 2. *Sea $A = (V, E')$ el árbol de difusión de una gráfica $G = (V, E)$ obtenido de la ejecución del algoritmo *AGDM*. Supongamos que se ejecuta el algoritmo *ADM* con A como parámetro para difundir un mensaje M y que $v_i \in V$ recibe en determinado momento a M . Entonces, v_i será capaz de comunicarse con sus vecinos en A y enviar una copia de M a cada uno de ellos.*

Demostración. Para demostrar que v_i puede comunicarse con todos sus vecinos en A , basta con demostrar que es capaz de comunicarse con los vecinos que tiene por alguno de sus sectores. Supongamos, sin pérdida de generalidad, que v_i transmite por el sector s .

Sabemos que el tiempo $d_{v_i s}$ corresponde al mayor valor de k_{v_j} con $v_j \in N_s(v_i)$. Entonces se tiene que:

$$d_{v_i s} = \max\{k_{v_j}\}; v_j \in N_s(v_i), \quad (3.1)$$

por lo que,

$$d_{v_i s} \geq k_{v_j}; \text{ para cualquier } v_j \in N_s(v_i). \quad (3.2)$$

Es decir, el tiempo que el nodo v_i mantiene la transmisión sobre su sector s permite que cada uno de sus nodos vecinos en $N_s(v_i)$ transmita al menos una vez por cada uno de sus sectores, ya que con el algoritmo ADM se asegura que los nodos vecinos de v_i no reciben a M desde algún otro nodo que no sea él, entonces cada uno mantiene la transmisión durante una unidad de tiempo en cada uno de sus sectores hasta recibir el mensaje. Por lo tanto, en algún momento el nodo v_i podrá comunicarse con cada uno de sus vecinos en $N_s(v_i)$ y les enviará una copia de M . \square

3.3. Ejemplo de difusión

En esta sección se ejemplifica la difusión de un mensaje haciendo uso de los algoritmos de la Sección 3.2. Durante la ejecución del algoritmo AGDM se mostrarán todos los conjuntos disponibles por iteración. Los nodos incluidos en cada conjunto candidato se colorean con un color distinto y los enlaces que sean incluidos en el árbol se marcan con rojo. Por otra parte, durante la ejecución del algoritmo ADM se analiza la difusión del mensaje en el árbol resultante.

La red de ejemplo consta de ocho nodos sensores y la gráfica G presentada en la Figura 3.5 la representa. Supongamos que la petición de difusión de un determinado mensaje M llega a v_0 . Al surgir esta petición comienza a ejecutarse el algoritmo AGDM con v_0 y G como parámetros. Al comienzo, el conjunto A contiene solamente al nodo v_0 y el conjunto R a todos los restantes. La Figura 3.5 muestra como se ve G en ese momento.

La Figura 3.6 muestra los conjuntos que están disponibles en C durante la primera iteración del ciclo. En este caso, el conjunto de la Figura 3.6a agrega más nodos al árbol. Por lo tanto, como se muestra en la Figura 3.7, se agregan los nodos v_1 y v_7 al conjunto A y se borran de R , además de incluir los enlaces (v_0, v_1) y (v_0, v_7) al árbol. El valor de la variable $d_{v_0,1}$ es 5, pues v_7 es el nodo con más sectores en el conjunto *conjuntoCandidato* $_{v_0}$. Este valor indica que el nodo v_0 mantendrá la transmisión durante cinco unidades de tiempo sobre su sector uno, que es el sector por el cual puede comunicarse con ambos nodos.

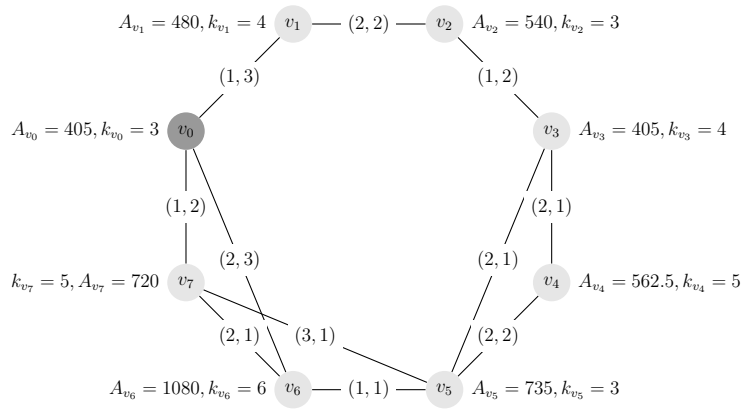
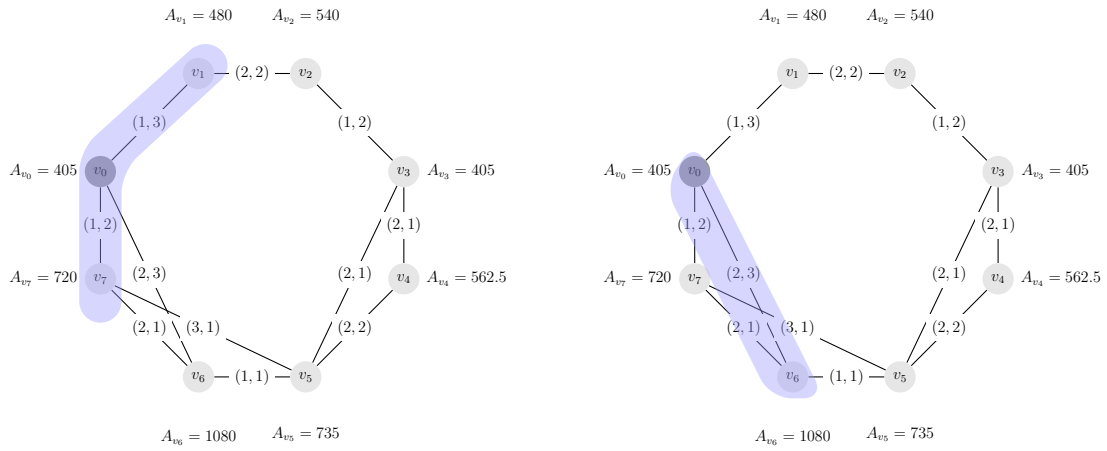


Figura 3.5: Petición de difusión de mensaje a v_0 . $A = \{v_0\}$ y $R = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$



(a) *conjuntoCandidato* $_{v_0}$. Consumo de 405 unidades de energía

(b) Consumo de 405 unidades de energía

Figura 3.6: Conjuntos disponibles en la primera iteración

En la ejecución de la segunda iteración los conjuntos disponibles en C tienen distintas cardinalidades. Como el algoritmo busca incluir el mayor número de nodos al árbol en cada iteración, el conjunto de la Figura 3.8c es el conjunto *conjuntoCandidato* $_{v_7}$ de esta iteración. Así que, los nodos v_5 y v_6 se agregan a A , se borran de R y los enlaces (v_7, v_5) y (v_7, v_6) son incluidos también en el árbol. La variable d_{v_7} toma el valor de 6, pues el nodo v_6 es el nodo con más sectores dentro del conjunto elegido. El estado del árbol puede observarse en la Figura 3.9.

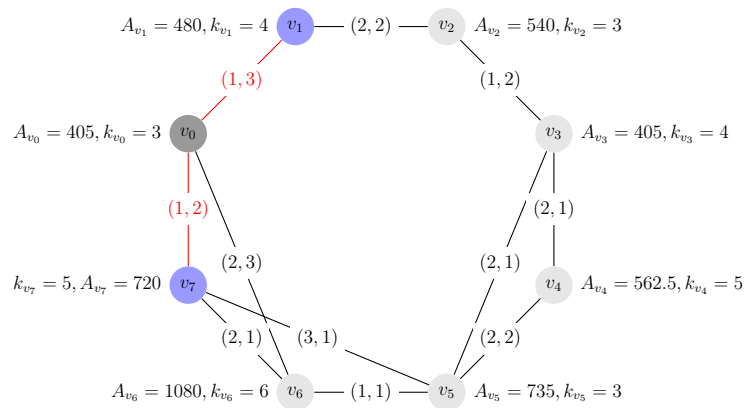
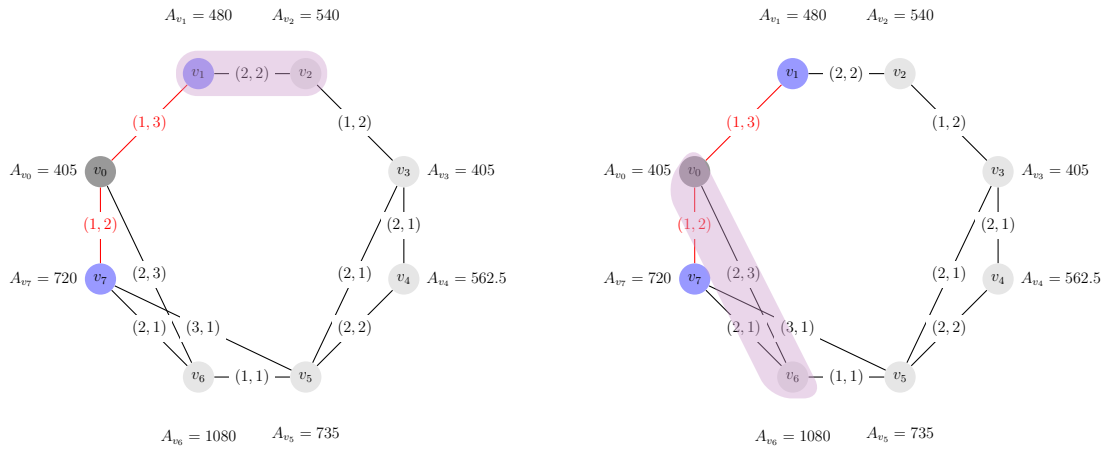
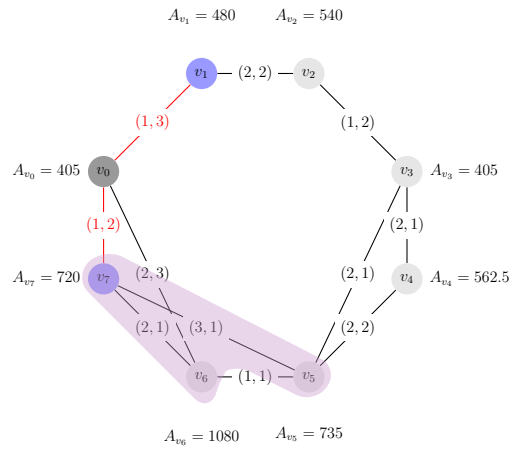


Figura 3.7: Estado del árbol antes de segunda iteración. $A = \{v_0, v_1, v_7\}$ y $R = \{v_2, v_3, v_4, v_5, v_6\}$



(a) Agrega un nodo con un consumo de 480 unidades de energía

(b) Agrega un nodo con un consumo de 405 unidades de energía



(c) *conjuntoCandidato*_{v7}. Agrega dos nodos con un consumo de 720 unidades de energía

Figura 3.8: Conjuntos disponibles en la segunda iteración

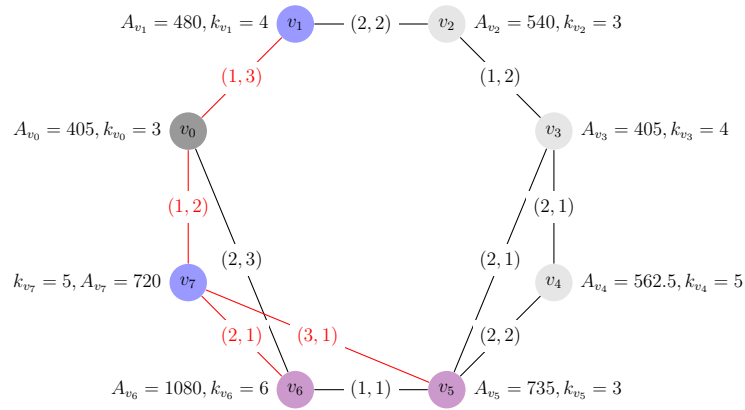
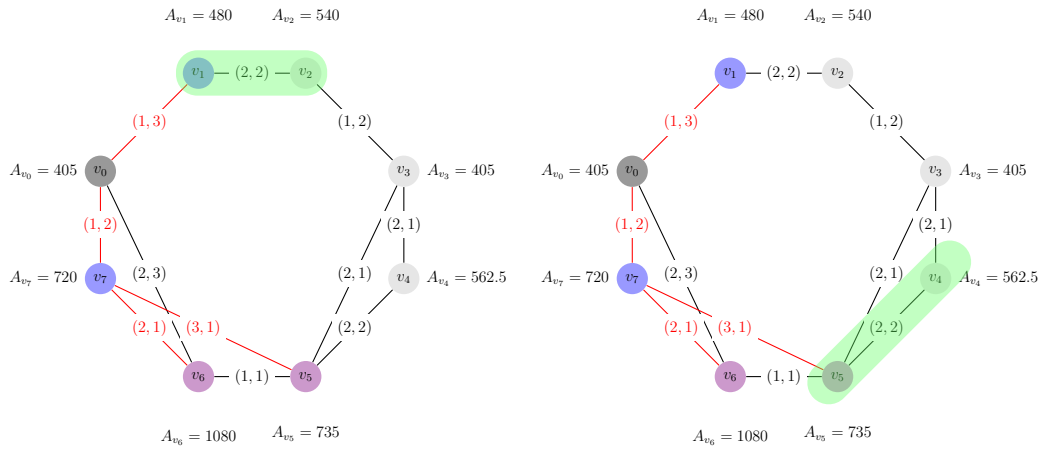
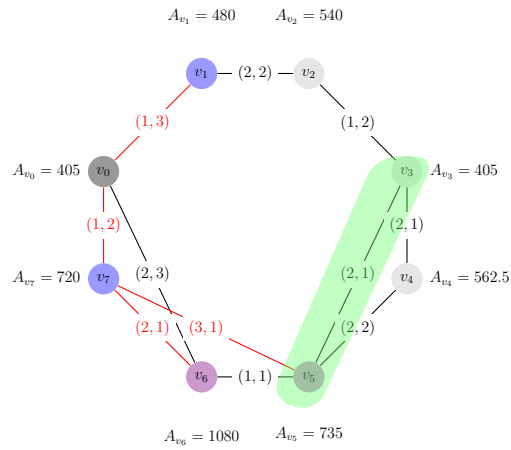


Figura 3.9: Estado del árbol antes de tercera iteración. $A = \{v_0, v_1, v_7, v_5, v_6\}$ y $R = \{v_2, v_3, v_4\}$

En la tercera iteración todos los conjuntos tienen la misma cardinalidad, de modo que el conjunto que resulta candidato es el de la Figura 3.10a, al ser el de menor consumo de energía. Como se ilustra en la Figura 3.11 se agrega el nodo v_2 a A y el enlace (v_1, v_2) se incluye en el árbol. El valor de la variable $d_{v_1,2}$ es 3, pues v_2 es el único nodo en el *conjuntoCandidato* $_{v_1}$.



(a) *conjuntoCandidato* $_{v_1}$. Agrega un nodo con un consumo de 480 unidades de energía
 (b) Agrega un nodo con un consumo de 735 unidades de energía



(c) Agrega un nodo con un consumo de 735 unidades de energía

Figura 3.10: Conjuntos disponibles en la tercera iteración

Durante la cuarta iteración el conjunto de la Figura 3.12c resulta candidato, pues es el conjunto con el menor gasto de energía. En esta iteración se agrega el nodo v_3 al árbol junto con el enlace (v_2, v_3) . El valor de la variable $d_{v_2 1}$ es 4.

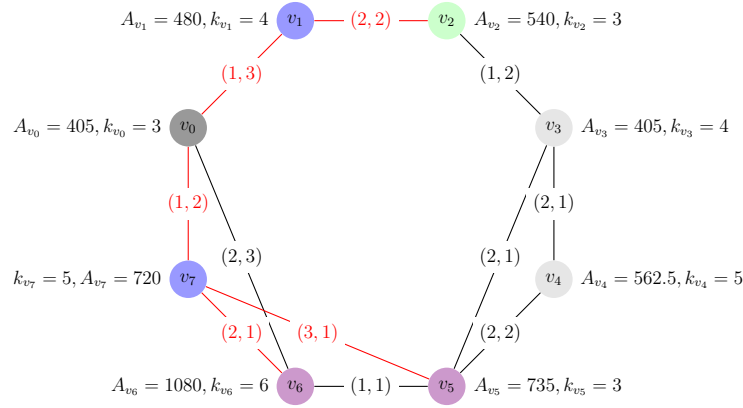
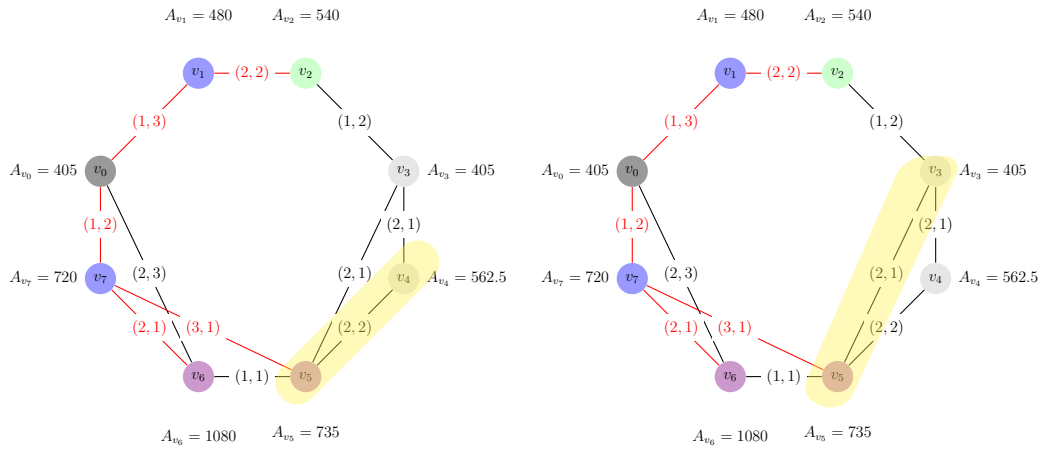


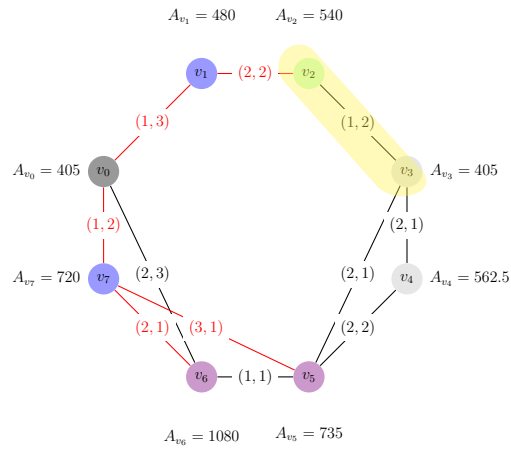
Figura 3.11: Estado del árbol antes de la cuarta iteración. $A = \{v_0, v_1, v_7, v_5, v_6, v_2\}$ y $R = \{v_3, v_4\}$

En la quinta iteración el conjunto de la Figura 3.14b es el próximo que se agregará al conjunto A. En realidad ambos conjuntos contienen solamente al nodo v_4 , pero se gasta menos energía al transmitir desde v_3 a v_4 . Entonces, el nodo v_4 se agrega al árbol junto con el enlace (v_3, v_4) . Además, el valor de la variable $d_{v_3 2}$ es 5.

Al terminar la quinta iteración el conjunto R queda vacío. En consecuencia el algoritmo termina su ejecución, dando como resultado un árbol de la gráfica G que incluye todos sus nodos y siete enlaces, representado en la Figura 3.15.



(a) Agrega un nodo con un consumo de 735 unidades de energía
 (b) Agrega un nodo con un consumo de 735 unidades de energía



(c) *conjuntoCandidato* $_{v_2}$. Consumo de 540 unidades de energía

Figura 3.12: Conjuntos disponibles en la cuarta iteración

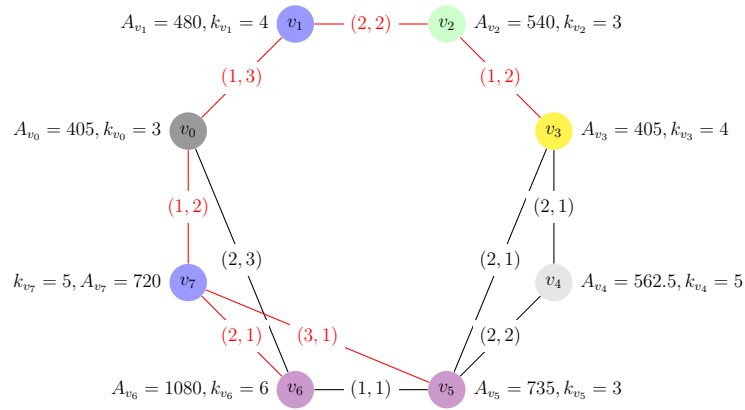
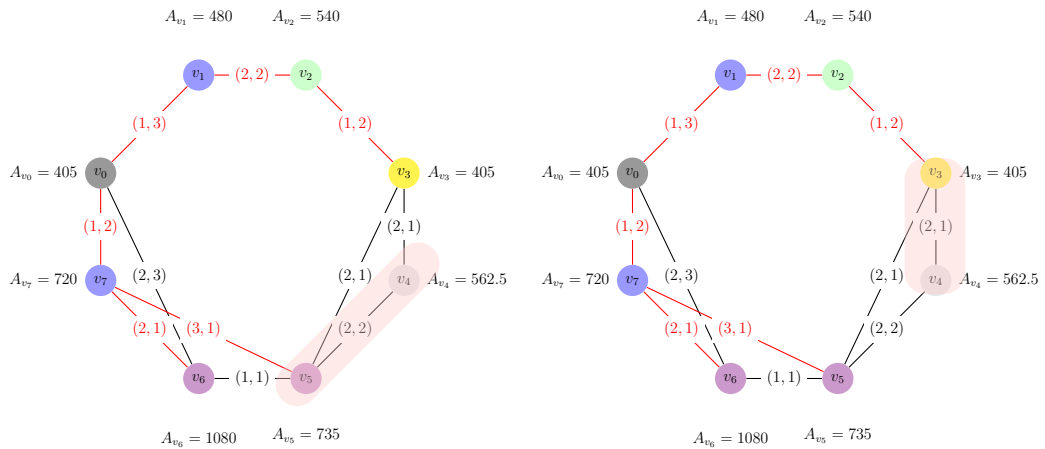


Figura 3.13: Estado del árbol antes de la quinta iteración. $A = \{v_0, v_1, v_7, v_5, v_6, v_2, v_3\}$ y $R = \{v_4\}$



(a) Agrega un nodo con un consumo de 735 unidades de energía
 (b) *conjunto* $Candidato_{v_3}$. Agrega un nodo con un consumo de 405 unidades de energía

Figura 3.14: Conjuntos disponibles en la quinta iteración

Para llevar a cabo la difusión del mensaje se ejecuta el algoritmo ADM con el árbol de la Figura 3.15 como parámetro. Cuando se inicia la ejecución, v_0 es el único nodo que posee el mensaje y el que se encargará de iniciar la difusión.

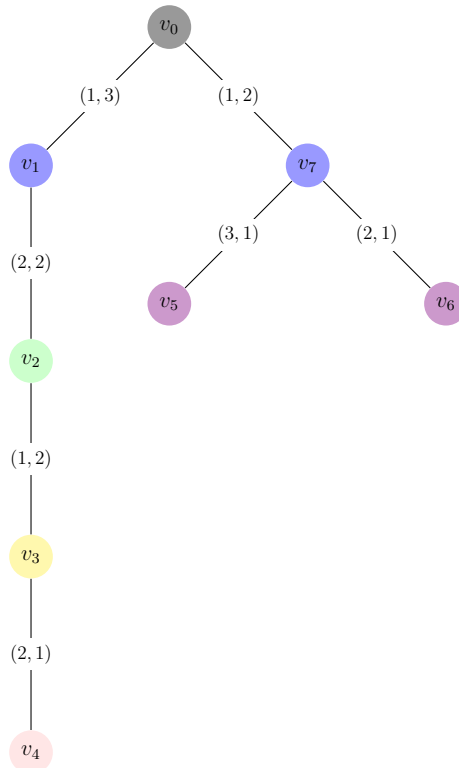
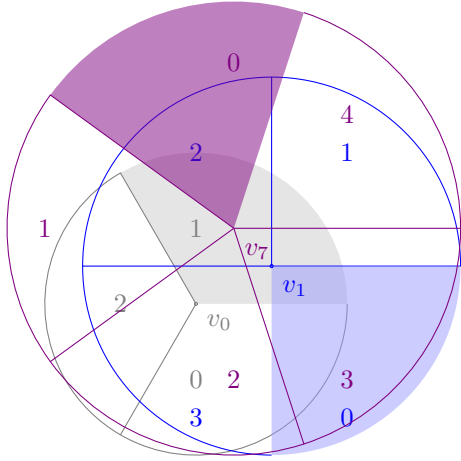


Figura 3.15: Árbol de difusión

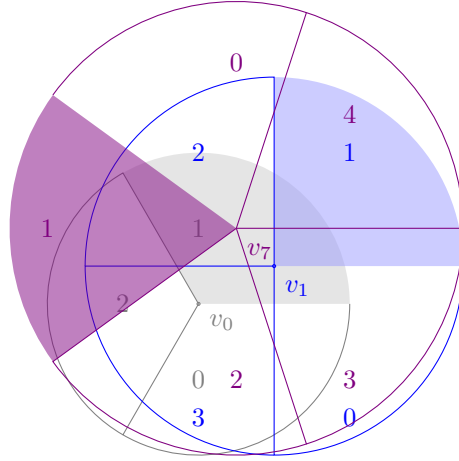
Para ejemplificar la ejecución del algoritmo ADM se mostrará cómo el nodo v_0 envía el mensaje a los nodos que tiene en el árbol, es decir, a los nodos v_1 y v_7 . Una vez que v_0 recibe el mensaje M comienza a transmitir por sus sectores que tienen vecinos, en este caso transmite solamente por su sector uno. En el ejemplo se usará una variable t que indica el tiempo transcurrido en la ejecución. Al comenzar la ejecución el valor de t es 1. Cuando surge la petición de difusión de M , el nodo v_0 gira su antena hacia el sector uno y comienza a transmitir energía durante $d_{v_0,1}$ unidades de tiempo, es decir, durante cinco unidades de tiempo.

Por otro lado, supongamos que todos los nodos restantes en la red comienzan a transmitir por su sector cero al tiempo $t = 1$. Entonces, mientras v_0 transmite por su sector uno, la antena de los nodos v_1 y v_7 transmitirá por cada uno de sus sectores comenzando por el sector cero. De acuerdo a la Figura 3.16a, al tiempo $t = 1$ el nodo v_0 transmite por el sector uno y los nodos v_1 y v_7 transmiten por su sector cero. Como se muestra en la Figura 3.16b, al tiempo $t = 2$ los nodos v_1 y v_7 giran su antena hacia el sector uno. Hasta ese momento ninguno de los nodos ha tenido la posición correcta para poder comunicarse con v_0 . Sin embargo, al tiempo $t = 3$, v_1 y v_7 giran su antena hacia el sector dos, por lo que v_7 adopta la posición adecuada para tener comunicación con v_0 . Esta situación puede observarse en la Figura 3.16c. En esta unidad de tiempo v_7 envía un mensaje que contiene su identificador y v_0 lo recibe e identifica a v_7 como su vecino en el árbol, así que le responde con una copia de M . En el tiempo $t = 4$ el nodo v_1 gira su antena hacia el sector tres, lo cual permite a v_1 alinear su antena con la antena de v_0 , tal como se muestra en la Figura 3.16d. El nodo v_0 recibe el mensaje de v_1 que contiene su identificador, lo identifica como su vecino en el árbol y le manda una copia de M . Cuando $t = 5$ el nodo v_0 ya no tiene vecinos en el árbol, así que detiene su ejecución del algoritmo ADM, aunque se seguirá ejecutando en los nodos que aún no entreguen a todos sus vecinos una copia de M .

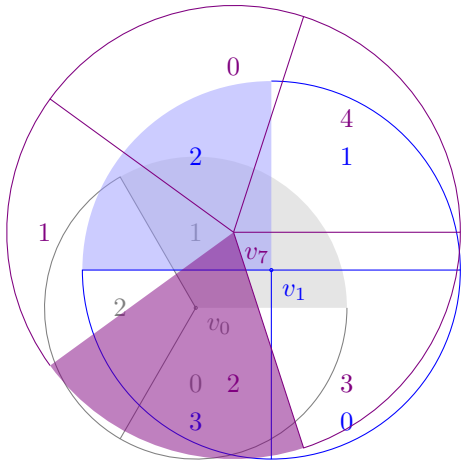
Cada nodo que recibe a M sigue el proceso anterior, de manera que cada uno entregará el mensaje M a sus vecinos. Entonces, en un tiempo finito el mensaje habrá sido entregado a todos los nodos de la red.



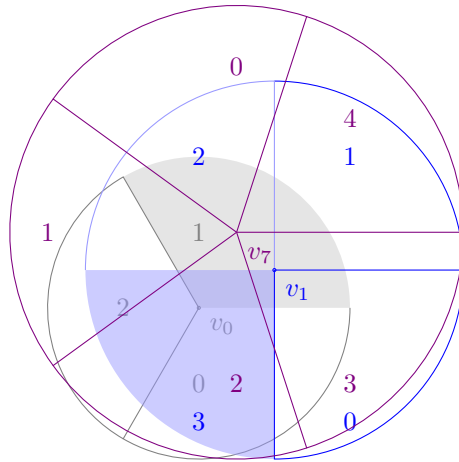
(a) $t = 1$. Nodos v_1 y v_7 comienzan a transmitir por su sector cero.



(b) $t = 2$. Nodos v_1 y v_7 giran su antena al sector uno.



(c) $t = 3$. Nodos v_1 y v_7 giran su antena al sector dos. Se comunican v_0 y v_7 .



(d) $t = 4$. Nodo v_1 gira su antena al sector tres. Se comunican v_0 y v_1 .

Figura 3.16: Ejemplo de la rotación de antenas en ADM

Conclusiones

En este texto se ha discutido sobre los beneficios que surgen al usar antenas direccionales en una red inalámbrica de sensores, debidos a la forma en que estas antenas transmiten energía. Entre estos beneficios se encuentran el reuso espacial, el aumento de conectividad y de la seguridad, así como el ahorro de energía. Esta última característica en particular es de las más importantes, ya que de ésta puede depender el tiempo de vida de la red, pues en muchos casos no se dispone de una infraestructura de energía y entonces los nodos deben sobrevivir con una cantidad limitada. A pesar de esta importante ventaja se mencionó que aparecen nuevos problemas a atacar, entre los más importantes está la comunicación, pues las condiciones para poder mandar o recibir datos de un nodo a otro se vuelven más complicadas al tener que posicionar de una manera específica la antena de cada uno de ellos. Entonces, es importante llevar a cabo el diseño de nuevos algoritmos que permiten la comunicación en la red.

Los algoritmos aquí propuestos para la difusión de mensajes en este tipo de redes sugieren una forma en la que la información que debe ser común para todos los nodos en la red pueda estar disponible para ellos. Se demostró teóricamente que al aplicar ambos algoritmos a una gráfica que representa a una red inalámbrica de sensores, se construye una solución para la difusión de un mensaje desde un nodo determinado hacia los otros nodos en la gráfica.

La difusión de mensajes es tan sólo uno de los problemas de redes que se deben resolver. Problemas como encontrar las trayectorias más cortas o las de menor costo son interesantes de analizar en este tipo de redes. Además de que proporcionar algoritmos que brinden una solución que lleve a cabo un ahorro energético, haría que el uso de antenas direccionales parezca aún más, la mejor opción a utilizar.

A pesar de que la red se considera un sistema distribuido, el algoritmo AGDM no se trata de un algoritmo distribuido, debido a que su definición sale del alcance de esta tesis, ya que son necesarios tanto un análisis más complejo como una teoría más completa. Sin embargo, el algoritmo ADM pretende ser un pequeño acercamiento a un algoritmo distribuido, puesto que se ejecuta en cada uno de los nodos de manera independiente. ADM no cumple con todos los criterios necesarios para considerarse un algoritmo distribuido diseñado correctamente para resolver el problema de difusión de mensajes en una red inalámbrica de sensores. Aunque se demuestra que se entrega el mensaje a cada nodo de la red, un aspecto importante que no se cubre es la *robustez*, es decir, el algoritmo no es capaz de actuar bajo un cambio en la red, pues gracias a que los nodos pueden estar bajo un esquema sin infraestructura energética, estos pueden quedarse sin fuente de energía y provocar que la topología de la red cambie. Entonces, el algoritmo debería ser capaz de detectar la situación, adaptarse y funcionar de nuevo de manera correcta.

Otro de los aspectos importantes a considerar es la *eficiencia* u *optimización* del algoritmo. El algoritmo se considera óptimo si envía información a través de las mejores trayectorias. En este caso, las trayectorias que se pretenden encontrar son aquéllas que permitan un ahorro energético para poder aumentar la vida de la red. Para ello, el algoritmo AGDM elige de los conjuntos candidatos disponibles en cada iteración de su ejecución, incluir en el árbol de difusión al conjunto con el mayor número de nodos y con el menor consumo de energía. La idea principal de escoger cada conjunto de la manera descrita es lograr enviar el mayor número posible de copias de un mensaje por el mismo consumo de energía. Sin embargo, con esta elección del conjunto no se logra tener las mejores trayectorias, ya que el consumo de energía también depende de la cantidad de tiempo que un nodo en la red mantiene la transmisión por los sectores que tienen vecinos en el árbol. De modo que, mediante el sencillo ejemplo de la Figura 3.18 se puede observar que para el ejemplo de la Sección 3.3, representado en la Figura 3.17, el árbol de difusión podría tener otra topología y que los tiempos de transmisión podrían tomar otros valores para que al ejecutar el algoritmo ADM el consumo de energía sea menor. En la figura se indica la cantidad de energía total que se consume en la difusión del mensaje por toda la red. Por lo tanto, el algoritmo ADM no resulta óptimo, pues las trayectorias que utiliza dependen del algoritmo AGDM. Así que es importante la definición de un algoritmo distribuido que

genere un árbol de difusión y que defina las mejores trayectorias para difundir un mensaje y así poder asegurar la eficiencia de ADM.

En ocasiones los nodos en la red no están sobre una superficie fija, pues precisamente la información que pretende ser recolectada depende del movimiento de aquéllo que es monitoreado, como por ejemplo, la ubicación de un automóvil, de una persona o de un animal. Entonces, es interesante desarrollar algoritmos de comunicación que tomen en cuenta esta característica de movilidad de los nodos en redes inalámbricas de sensores que hacen uso de antenas direccionales.

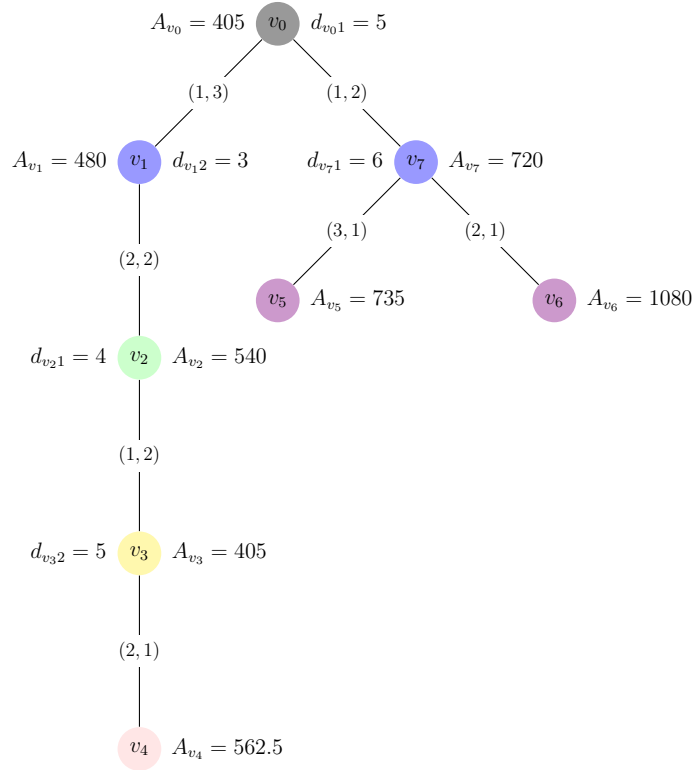


Figura 3.17: Ejemplo del árbol de difusión con tiempos de transmisión. Consumo total de energía : 11970 unidades.

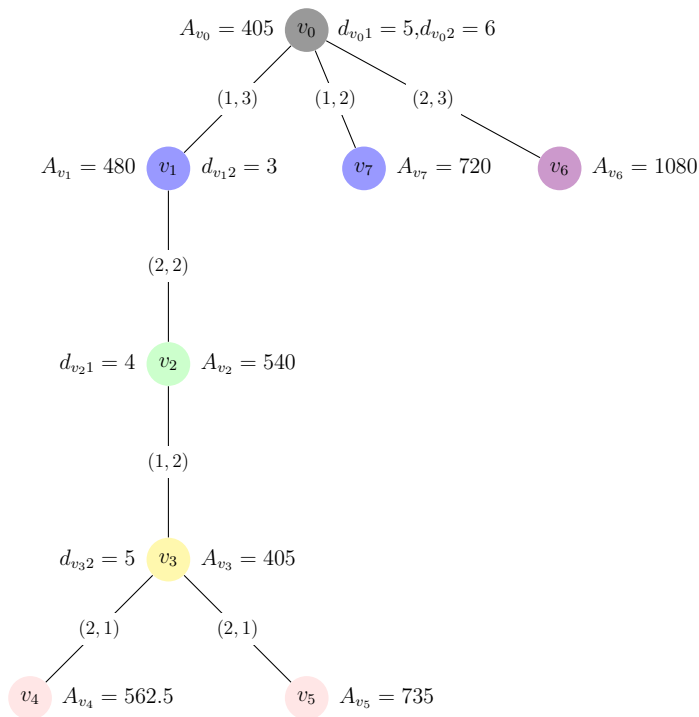


Figura 3.18: Ejemplo del árbol de difusión con diferente topología. Consumo total de energía: 10507.5 unidades.

Un problema también interesante es el diseño de algoritmos en donde los nodos puedan, de manera dinámica, ajustar el tamaño de sus sectores para maximizar el número de vecinos a los que pueden mandar un mensaje, o bien, considerar que un nodo posee más de una antena direccional.

Por último, realizar simulaciones sobre distintas topologías de red de los algoritmos propuestos en este texto es un trabajo útil para realizar, debido a que los resultados podrían ser comparados con los de otros algoritmos que han sido implementados en trabajos anteriores.

Glosario

ADM Algoritmo de Difusión de Mensajes. 28, 29, 49, 50

AGDM Algoritmo Generador del árbol de Difusión de Mensajes. 28,29,32,50

ARA Antenna Rotation Algorithm.
Algoritmo de Rotación de Antena. 25–27

CT1 Connectivity Test 1.
Prueba de Conectividad 1. 16, 17

CT2 Connectivity Test 2.
Prueba de Conectividad 2. 17

CT3 Connectivity Test 3.
Prueba de Conectividad 3. 17

CT4 Connectivity Test 4.
Prueba de Conectividad 4. 17, 18

CT5 Connectivity Test 5.
Prueba de Conectividad 5. 17, 18

GLR Generic Link Reversal algorithm.
Algoritmo de Inversión de Enlaces Genérico. 20

LRME Link Reversal Mutual Exclusion.
Algoritmo de Inversión de Enlaces para la Exclusión Mutua. 21

- MH** Minimum Hop path.
Trayectoria con el Mínimo número de Enlaces. 18
- PI** Propagation of Information.
Propagación de Información. 16
- PIF** Propagation of Information with Feedback.
Propagación de Información con Retroalimentación. 16, 17
- RARA** Randomized Antenna Rotation Algorithm.
Algoritmo de Rotación Aleatoria de Antena. 26, 27
- RSRMA** Random Selection Rotation Mechanism Algorithm.
Algoritmo de Selección Aleatoria de la Rotación de Antena. 26, 27

Apéndice A

Teoría de gráficas

En este apéndice se proporcionan algunas definiciones básicas de la teoría de gráficas extraídas del texto de los autores Miranda et al. [7]. Se comienza definiendo el concepto de gráfica, continuando con algunas propiedades de sus componentes y se finaliza con la definición de dos tipos de gráficas importantes. Al introducir estos términos se pretende mostrar la notación que se adoptará en el texto para referirse a las gráficas o a sus componentes, además de ofrecer al lector las herramientas necesarias para la comprensión del texto.

Una *gráfica* $G = (V, E)$ se define como una pareja que consiste de un conjunto no vacío de puntos V , llamados *vértices* o *nodos* y un segundo conjunto E , subconjunto de $V \times V$, de *aristas* o *enlaces* que corresponden a parejas no ordenadas de vértices (u, v) , donde los vértices u y v son los extremos de una arista. Los vértices y las aristas pueden caracterizarse de distintas maneras, por ejemplo, a dos vértices u y v en la gráfica se les dice *vértices adyacentes* si la arista $(u, v) \in E$, es decir, si existe una arista en E cuyos extremos son u y v . Otra característica es que una arista es *incidente* en un vértice u si u es uno de los extremos de la arista. Por último, se define el *grado* de un vértice, denotado como $\text{grado}(v)$, como el número de aristas incidentes en él.

Para ejemplificar los conceptos anteriores, en la Figura A.1 se muestra un ejemplo de una gráfica con tres vértices y dos aristas, las cuales son representadas con una línea que conecta a sus dos vértices extremos. Además los vértices v_0 y v_1 son adyacentes, pero v_0 y v_2 no lo son y la arista (v_0, v_1) es incidente en el vértice v_0 , así como en el vértice v_1 . Adicionalmente, el grado del vértice v_0 es 1, mientras que el del vértice v_1 es 2.

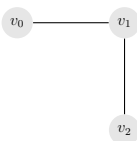


Figura A.1: Ejemplo de una gráfica G

Otro concepto importante relacionado con el recorrido de las gráficas es el de *camino*. Un camino en una gráfica es una sucesión de vértices intercalados con aristas, donde el primer y último elemento de la sucesión son vértices. Si el primer vértice de la sucesión es u y el último es v , decimos que tenemos un camino de u a v . Los caminos pueden ser de distintos tipos, por ejemplo, un camino entre dos vértices u y v en el que no se repiten vértices ni aristas se le conoce como *trayectoria*. Por otra parte, se conoce como *ciclo* a un camino en el cual el primer vértice y el último son el mismo.

En la Figura A.2 se ejemplifica con color gris una trayectoria del vértice v_0 al v_5 en una gráfica, que pasa también por los vértices v_1 , v_2 y v_3 , así como por las aristas que los conectan. Asimismo, el camino $v_1 - v_2 - v_3 - v_4 - v_1$ de la misma figura es un ejemplo de un ciclo.

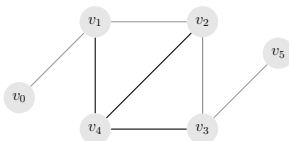


Figura A.2: Ejemplo de un camino en una gráfica

En ocasiones resulta de interés saber si existe algún camino entre los vértices de una gráfica, por lo que la siguiente definición resulta de gran utilidad. Se dice que una gráfica $G = (V, E)$ es *conexa* si para cualesquiera dos vértices en V , u y v , existe un camino entre ellos. La Figura A.3 muestra un ejemplo de una gráfica conexa y otra no conexa. En la gráfica no conexa es claro que no hay un camino de v_0 a v_3 .

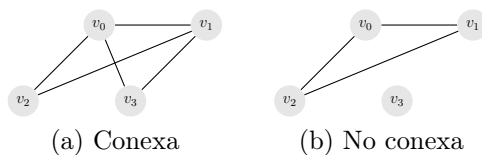


Figura A.3: Ejemplo de una gráfica conexa y una no conexa

La siguiente propiedad puede utilizarse para modelar ciertos problemas haciendo uso de gráficas. Esta propiedad es conocida como *coloración*, que es el número de colores que se requieren para colorear a una gráfica, de forma tal que cualquier par de vértices adyacentes reciban distinto color. En la Figura A.4 se muestra la coloración de una gráfica con cuatro vértices y dos colores.

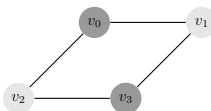


Figura A.4: Coloración de una gráfica

Las últimas definiciones introducen dos caracterizaciones de una gráfica. La primera introduce direcciones a las aristas en una gráfica y se conoce como *gráfica dirigida*, la cual se define como una gráfica donde los elementos del conjunto E ahora son pares ordenados. De modo que, $(u, v) \neq (v, u)$. En la Figura A.5 se ilustra una gráfica dirigida con cuatro vértices y cuatro aristas.

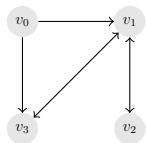


Figura A.5: Ejemplo de una gráfica dirigida

La segunda caracterización se llama *árbol*, que es una gráfica que no contiene ciclos y es conexa. La Figura A.6 muestra un ejemplo de este tipo de gráficas. Se observa que no existe un camino que comience y termine en un mismo vértice, sin embargo, se puede ir de un vértice a otro.

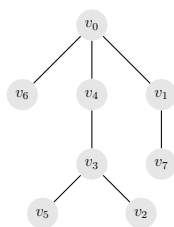


Figura A.6: Ejemplo de un árbol

Dos conceptos sobre un árbol necesitan ser definidos, sin embargo sus definiciones no se encuentran en la fuente mencionada anteriormente. Estos conceptos se refieren a elementos de un árbol que se denominan *hoja* y *rama*. Una *hoja* es un nodo en el árbol que tiene grado 1 y una *rama* es un camino desde el nodo raíz a una hoja del árbol. En la Figura A.6 el nodo v_2 representa una hoja y el camino $v_0 - v_1 - v_7$ una rama.

Apéndice B

Teoría de redes

En este apéndice se proporcionan algunas definiciones básicas de las redes de computadoras. Se comienza por una definición general de una red y después se define a una red de computadoras, así como algunos de sus componentes. Posteriormente se proveen algunos conceptos importantes para la creación de una red.

Gallo et al. [4] definen a una *red* como un grupo de sistemas conectados con el propósito de compartir información. Un tipo de red es la *red de computadoras*, que es definida por los autores Gallo et al. [3] como una colección de computadoras y otros dispositivos, llamados comúnmente *nodos*, que usan un protocolo de red para compartir recursos entre ellos a través de un medio de red.

Para poder compartir información o recibir un servicio, los nodos de la red deben estar conectados entre sí de alguna manera y ser capaces de entenderse, lo cual implica seguir ciertas reglas de comunicación. En la Figura B.1 se muestra un ejemplo de una red de sensores que es también una red de computadoras.¹

Un *medio de red* se define como el ambiente físico usado para conectar los miembros de la red. Estos medios se presentan en dos amplias categorías: cables e inalámbricos. Ejemplos de medio cable son el cable de par trenzado, el cable coaxial y el cable de fibra óptica. Ejemplos del medio inalámbrico son las ondas de radio (incluidas las microondas y la comunicación por satélite) y

¹<http://fiji.eecs.harvard.edu/Volcano>. Consultada en septiembre, 2013

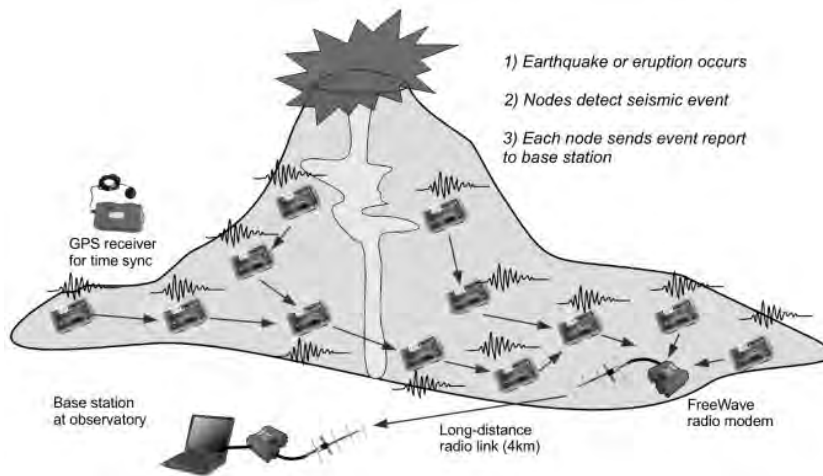


Figura B.1: Red de sensores

la radiación infrarroja, [3]. Sin embargo, aunque los nodos estén conectados mediante un medio, se necesita que usen el mismo lenguaje. Este lenguaje se llama *protocolo de red* y define tanto el vocabulario como las reglas de la comunicación de los datos. Este protocolo también define el formato de los datos a intercambiar, cómo se mantendrá la integridad de estos y cómo se reciben y envían entre los nodos, [4].

Continuando con los tipos de medios de una red, se explica con un poco más de detalle la manera en la que se llevan a cabo las transmisiones y recepciones inalámbricas. Para este medio no basado en cables, la transmisión y recepción se realizan por medio de antenas. Este principio lo esclarece Tanenbaum [10] cuando explica la transmisión inalámbrica. Básicamente lo que sucede es que cuando los electrones se mueven crean ondas electromagnéticas que pueden propagarse al espacio libre. De modo que, al conectar una antena del tamaño apropiado a un circuito eléctrico las ondas pueden difundirse de manera eficiente y ser captadas por un receptor a cierta distancia.

Como Stallings [9] explica, existen básicamente dos tipos de configuración para la transmisión inalámbrica: direccional y omnidireccional. Para la direccional la antena transmite por un rayo electromagnético; así que las antenas emisoras y receptoras deben estar alineadas correctamente. En el caso omnidireccional, la señal transmitida se propaga en todas las direcciones y

puede recibirse por varias antenas.

La Figura B.2 muestra el espectro electromagnético con las frecuencias de los distintos tipos de ondas y las aplicaciones que tienen en comunicaciones de datos.²

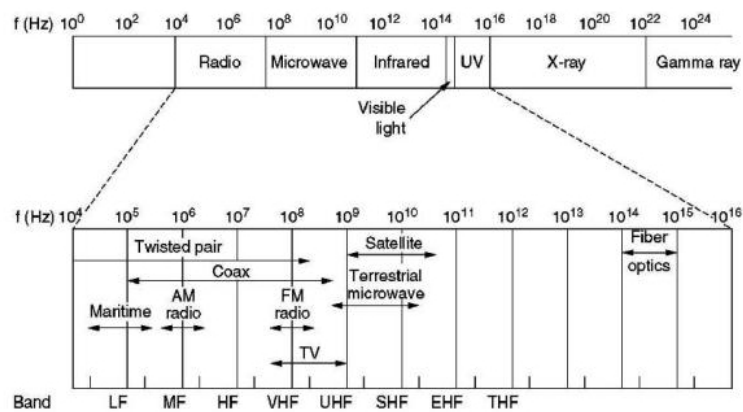


Figura B.2: Espectro electromagnético

Otra característica importante de una red es su *topología*, que los autores Gallo et al. [3] definen como el diseño básico de una red y en la que se detalla cómo están conectados los componentes claves de ella. En pocas palabras, la topología es la forma de una red. En la Figura B.3 se muestran algunas de las topologías conocidas.

²<http://tutorial.galeon.com/inalambrico.htm>. Consultada en septiembre, 2013

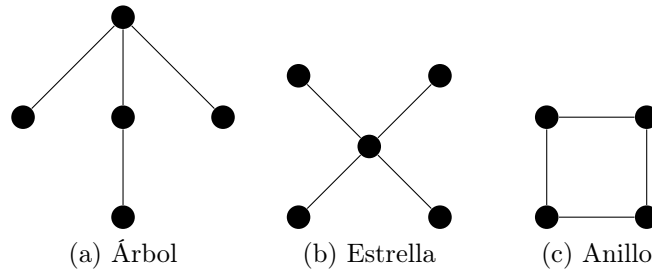


Figura B.3: Ejemplos de topologías de red

A continuación se proporcionan otros conceptos importantes y cuyas definiciones también se basan en el libro de Gallo et al. [3]. Estos conceptos son direccionamiento, enrutamiento, protocolo de enrutamiento y algoritmo de enrutamiento.

El *direccionamiento* describe cómo se asigna una única dirección a un nodo para que otros sean capaces de localizarlo e identificarlo. Relacionado al direccionamiento, se encuentra el *enrutamiento*, que describe la manera en que los datos son transferidos de un nodo a otro a través de la red. El enrutamiento involucra determinar el camino que un mensaje toma cuando viaja de un nodo fuente a un nodo destino.

El camino que un mensaje toma a través de una red, del nodo fuente al destino, está en función de los *protocolos de enrutamiento*, que tienen como funciones primarias: determinar el mejor camino que un mensaje debe tomar cuando viaja a través de una red y mantener las tablas de enrutamiento que contienen información acerca de la topología de la red. Estos protocolos se basan en *algoritmos de enrutamiento* que calculan el camino de menor costo de la fuente al destino y que son los responsables de decidir sobre qué línea de salida debe colocarse un mensaje que llega a un nodo fuente.

Como explica Tanenbaum [10], en ocasiones se necesita enviar mensajes a varios otros nodos o a todos los nodos en la red. El envío simultáneo de un mensaje a todos los destinos se llama *difusión* y hay distintos algoritmos de enrutamiento que la llevan a cabo. Sin embargo, la manera en que la difusión se realiza depende, entre otras cosas, de los recursos que tenga una red. Así que, dependiendo de los recursos disponibles deben diseñarse algoritmos de enrutamiento para la difusión de mensajes.

Bibliografía

- [1] T. M. Chiwewe, A Distributed Topology Control Technique for Low Interference and Energy Efficiency in Wireless Sensor Networks. Proyecto Fin de Carrera, University of Pretoria, 2010.
- [2] J. Du, E. Kranakis, O. Ponce y S. Rajsbaum, Neighbor Discovery in a Sensor Network with Directional Antennae. *Algorithms for Sensor Systems, Lecture Notes in Computer Science*, vol. 7111, Springer Berlin Heidelberg, 2012, 57–71.
- [3] M. A. Gallo y W. M. Hancock, *Comunicación entre computadoras y tecnologías de redes*. México: Thomson, 2002, 1^a ed.
- [4] M. A. Gallo y W. M. Hancock, *Networking Explained*. United States of America: Digital Press, 2002, 1^a ed.
- [5] D. Li, Z. Li y L. Liu, Energy Efficient Broadcast Routing in Ad Hoc Sensor Networks with Directional Antennas. *Wireless Algorithms, Systems, and Applications*, (Eds.) Y. Li, D. Huynh, S. Das y D.-Z. Du, *Lecture Notes in Computer Science*, vol. 5258, Springer Berlin Heidelberg, 2008, 29–39.
- [6] G. Pei, M. Albuquerque, J. Kim, D. Nast y P. Norris, A neighbor discovery protocol for directional antenna networks. *Military Communications Conference, 2005. MILCOM 2005. IEEE*, vol. 1, 2005, 487–492.
- [7] F. E. M. Perea y E. V. Gurovich, *Matemáticas Discretas*. México, Distrito Federal: Las prensas de ciencias, 2010, 1^a ed.

- [8] A. Segall, Distributed network protocols. *Information Theory, IEEE Transactions on*, vol. 29, 1983, 23–35.
- [9] W. Stallings, *Data and computer communications*. United States of America: Prentice Hall, 2000, 6^a ed.
- [10] A. S. Tanenbaum, *Redes de computadoras*. Amsterdam, The Netherlands: Pearson, 2003, 4^a ed.
- [11] G. Tel, *Introduction to Distributed Algorithms*. Cambridge, 2000, 2^a ed.
- [12] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay y W. Hong, A macroscope in the redwoods. *Proceedings of the 3rd international conference on Embedded networked sensor systems*, SenSys '05, New York, NY, USA: ACM, 2005, 51–63.
- [13] S. Vasudevan, J. Kurose y D. Towsley, On neighbor discovery in wireless networks with directional antennas. *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, 2005, 2502–2512.
- [14] J. L. Welch y J. E. Walter, Link Reversal Algorithms. *Synthesis Lectures on Distributed Computing Theory*, vol. 2, 2011, 1–103.
- [15] Y. Xu, G. Chen, J. Ford y F. Makedon, Detecting Wormhole Attacks in Wireless Sensor Networks. *Critical Infrastructure Protection*, (Eds.) E. Goetz y S. Shenoi, *IFIP International Federation for Information Processing*, vol. 253, Springer US, 2007, 267–279.
- [16] J. Yick, B. Mukherjee y D. Ghosal, Wireless sensor network survey. *Computer Networks*, vol. 52, 2008, 2292 – 2330.
- [17] P. Zhang, C. M. Sadler, S. A. Lyon y M. Martonosi, Hardware design experiences in ZebraNet. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, New York, NY, USA: ACM, 2004, 227–238.