

**Informe Final de la Tesis de Profundización  
de Maestría: Integración de algoritmos  
orientados a la auto-curación de las redes  
inalámbricas de sensores**

**Sergio Andrés Díaz Salas**

Director de Tesis: Diego Méndez Chaves

Tesis presentada a la Pontificia Universidad Javeriana  
como requisito parcial para optar por el título de  
Magister en Ingeniería Electrónica

Mayo 12 2014

# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Marco Teórico</b>	<b>5</b>
2.1. Generación de Topología . . . . .	5
2.1.1. Clasificación de Protocolos de Enrutamiento . . . . .	6
2.1.2. Collection Tree Protocol (CTP) . . . . .	7
2.2. Selección de un Subconjunto de Nodos . . . . .	9
2.2.1. Creación de Clusters con Kmeans . . . . .	9
2.2.2. Particle Swarm Optimization (PSO) . . . . .	10
2.2.3. Métricas para la Selección de un Subconjunto de Nodos . . . . .	12
2.3. Generación de Caminos Disyuntos . . . . .	15
2.3.1. Algunos Autores Representativos . . . . .	15
2.3.2. N-to-1 Multipath Routing . . . . .	16
2.4. Generación de Mapa de Pérdida de Riesgo de Conectividad . . . . .	18
2.5. Matrices de comunicación asimétricas . . . . .	19
<b>3. Especificaciones</b>	<b>22</b>
3.1. Diagrama de Bloques DACA . . . . .	22
3.2. Diagrama de Bloques Castalia . . . . .	24
3.3. Especificaciones de DACA por módulo . . . . .	25
<b>4. Desarrollo Teórico</b>	<b>27</b>
4.1. Construcción de un árbol de comunicación inicial . . . . .	28
4.2. Creación de Clusters . . . . .	29
4.3. Optimización Multi-Objetivo . . . . .	31
4.4. Creación de Caminos Disyuntos . . . . .	33
<b>5. Análisis de Resultados</b>	<b>35</b>
5.1. Parámetros de la Simulación . . . . .	35
5.1.1. Capa de Aplicación . . . . .	35

5.1.2.	Capa de Comunicación . . . . .	36
5.1.3.	Manejo de Recursos . . . . .	39
5.2.	Métricas de la Simulación . . . . .	40
5.2.1.	Número de saltos hacia el sink . . . . .	40
5.2.2.	Tiempo de Vida de la Red: Cantidad de Nodos Activos e Inactivos Vivos . .	41
5.2.3.	Porcentaje de Cobertura . . . . .	41
5.2.4.	Mapa de Riesgo de Pérdida de Conectividad . . . . .	41
5.3.	Resultados de la Simulación . . . . .	41
5.3.1.	Generación de Clusters, Árboles y Caminos Disjuntos . . . . .	42
5.3.2.	Comparación entre DACA y CTP en Cobertura y Tiempo de Vida . . . . .	48
5.3.3.	Intercambio entre Cobertura y Tiempo de vida de la red . . . . .	52
5.3.4.	Mapa Riesgo de Pérdida de Conectividad . . . . .	53
<b>6.</b>	<b>Conclusiones</b>	<b>57</b>

# Índice de figuras

1.1. Diagrama General de las Redes de Sensores Inalámbricos . . . . .	2
1.2. Nodos . . . . .	3
1.3. Diagrama de Bloques General . . . . .	4
2.1. Módulos del Collection Tree Protocol . . . . .	8
2.2. Cluster . . . . .	10
2.3. Algoritmo K-means . . . . .	11
2.4. Creación de diagramas de voronoi con el esquema DASS . . . . .	14
2.5. Caminos Trenzados para Soportar la Falla de Algunos Nodos . . . . .	16
2.6. Árbol con Caminos Disyuntos . . . . .	18
2.7. Conteo de nodos . . . . .	19
2.8. Interpolación de Kriging . . . . .	20
2.9. Distancia máxima de transmisión a diferentes potencias . . . . .	21
3.1. Diagrama de bloques de la técnica DACA . . . . .	23
3.2. Estructura del nodo en Castalia . . . . .	24
3.3. Implementación de DACA en Castalia . . . . .	25
3.4. Variables de entrada y salida del árbol de comunicación inicial . . . . .	26
3.5. Variables de entrada y salida del MOO . . . . .	26
3.6. Variables de entrada y salida de la creación de clusters . . . . .	26
3.7. Variables de entrada y salida de la creación de caminos disyuntos . . . . .	26
4.1. Algoritmo Kmeans . . . . .	31
4.2. Árbol con Caminos Disyuntos . . . . .	34
5.1. Distancia máxima de transmisión a diferentes potencias . . . . .	38
5.2. Kmeans N-Iteration para $R_{sensado} = 5$ metros . . . . .	44
5.3. Árbol y caminos disyuntos para $R_{sensado} = 5$ metros . . . . .	45
5.4. Kmeans N-Iteration para $R_{sensado} = 10$ metros . . . . .	45
5.5. Árbol y Caminos Disyuntos para $R_{sensado} = 10$ metros . . . . .	46

5.6. Kmeans N-Iteration para $R_{sensado} = 15$ metros . . . . .	47
5.7. Árbol y Caminos Disjuntos para $R_{sensado} = 15$ metros . . . . .	47
5.8. Número de saltos según la potencia y el radio de sensado . . . . .	48
5.9. Comparación de nodos activos e inactivos entre DACA y CTP para $R_{sensado} = 5$ . .	49
5.10. Comparación de cobertura entre DACA y CTP para $R_{sensado} = 5$ . . . . .	50
5.11. Comparación nodos activos e inactivos entre DACA y CTP en la red para $R_{sensado} = 10$	51
5.12. Comparación de cobertura entre DACA y CTP para $R_{sensado} = 10$ . . . . .	51
5.13. Comparación nodos activos e inactivos entre DACA y CTP en la red para $R_{sensado} = 15$	52
5.14. Comparación de cobertura entre DACA y CTP para $R_{sensado} = 15$ . . . . .	53
5.15. Granularidad con $R_{sensado} \geq 15$ . . . . .	54
5.16. Granularidad con $R_{sensado} \leq 5$ . . . . .	54
5.17. Intercambio entre el tiempo de vida de la red y el porcentaje de cobertura. . . . .	55
5.18. Mapa Riesgo de Pérdida de Conectividad para DACA . . . . .	55
5.19. Mapa Riesgo de Pérdida de Conectividad para CTP . . . . .	56



# Capítulo 1

## Introducción

Este informe final muestra los resultados obtenidos durante en el desarrollo de la tesis de maestría cuyo título es Integración de Algoritmos Orientados a la Auto-Curación de las Redes Inalámbricas de Sensores. Este capítulo presenta de forma general los conceptos básicos de Redes de Sensores Inalámbricos (WSN) y muestra de forma resumida un diagrama de bloques del trabajo de tesis.

Las Redes de Sensores Inalámbricos (WSN) consisten de nodos de sensores distribuidos en un área geográfica para monitorear fenómenos físicos [1], son empleadas en aplicaciones militares y monitoreo ambiental. Los nodos tienen limitaciones en energía y en capacidad de procesamiento. Para superar estas limitaciones existe colaboración entre nodos, aumentando así, la robustez y tiempo de vida de la red. La colaboración inicia cuando los múltiples nodos son redundantes, lo que permite usar solamente algunos para medir la variable de interés y así disminuir el gasto de energía. En algunas aplicaciones la energía se puede renovar a través paneles solares u otro medio, pero no es el caso general, por lo tanto la prolongación del tiempo de vida de la red, a través del uso razonable de la energía, es un tema clave en las Redes de sensores inalámbricos [2].

Una característica interesante de las redes de sensores inalámbricos es que éstas se vuelven más robustas entre más nodos tenga [3], al contrario de otras redes de comunicaciones, debido a la naturaleza de colaboración entre nodos. Adicionalmente dado que las WSN se auto-curan , se auto-configuran , y se programan para consumir la menor cantidad de energía, son redes que no necesitan constante intervención humana.

Las características más apreciables de las WSN son las siguientes [4]: Escalabilidad: Dado que las WSN tienen nodos distribuidos densamente en un área reducida se presenta el problema de generar algoritmos de construcción de la red y enrutamiento de mensajes altamente escalables. Autonomía: Las WSN se emplean en lugares de difícil acceso o donde no es deseable la interven-

ción humana, por ejemplo un campo de batalla; esta característica hace que las WSN deban ser autónomas garantizando un tiempo de vida de la red que exceda varios meses o incluso años. Topología: Durante el tiempo de vida de la red su topología puede cambiar debido a movilidad o falla de los nodos, cambios en el canal de comunicación o cambios generados por el administrador de la red. Por lo anterior, una configuración manual de la topología de red sería desgastante, por ello se prefiere la auto-organización de los nodos. Volumen de datos: Los datos generados en las WSN se caracterizan por ser esporádicos y cortos, básicamente corresponden a peticiones del cliente o un bajo número de eventos.

Un diagrama básico de una red de sensores se observa en la figura 1.1. Una red WSN se compone de sensores distribuidos en un área geográfica, dichos sensores envían la información recolectada hacia el sumidero, el cual es el nodo principal donde se dirigen los datos obtenidos por los demás nodos. Para llegar al sumidero se tienen varias estrategias como el paradigma multisalto [1], el paradigma de árbol [5] y el paradigma de malla [6]. Estos paradigmas hacen un compromiso entre la energía de transmisión y recepción empleadas y el procesamiento de computación de la ruta que ahorra más energía. El sumidero tiene una mayor capacidad de computación y almacenamiento dado que se encarga de procesar los datos para evitar redundancias y enviarlos por medio de la red de tránsito hacia el servidor 1. Posteriormente los datos son enviados hacia un BS (Base Station) de algún ISP para enrutar los datos hacia el servidor 2 ubicado en la oficina del cliente a través de internet.

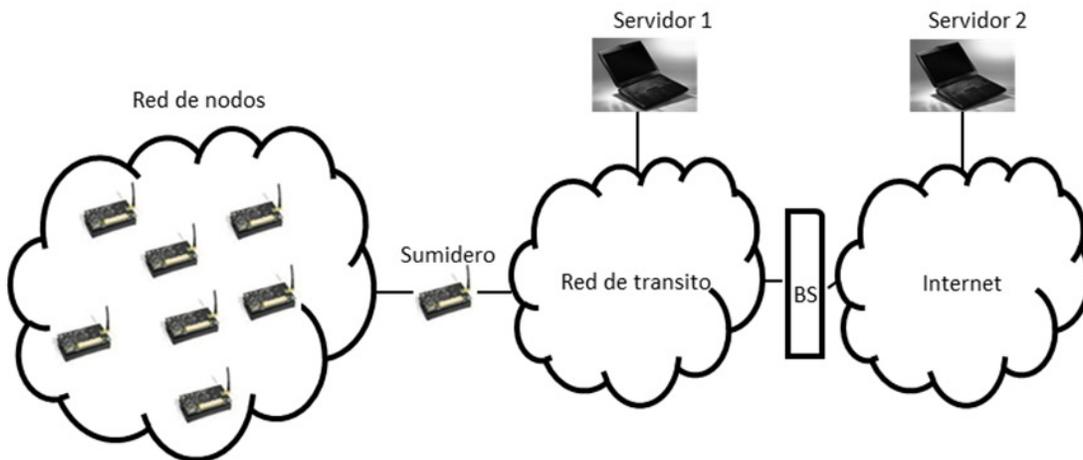


Figura 1.1: Diagrama General de las Redes de Sensores Inalámbricos

Un nodo se compone básicamente de: una batería, un subsistema de comunicaciones, un pro-

cesador limitado y un sensor. Cada nodo tiene un área de influencia que es el área alrededor de éste que es capaz de medir, figura 1.2. Adicionalmente, cada nodo tiene una potencia de transmisión y una potencia de recepción diferente, lo cual hace que los enlaces entre nodos no sean simétricos necesariamente. La cobertura es la suma de todas las áreas de influencia del conjunto de nodos y es el área que nos interesa caracterizar. Por ejemplo en la figura 1.2 se muestra que el nodo 3 alcanza al nodo 4 porque su potencia de transmisión sobrepone el área de recepción, de la misma manera el nodo 3 no alcanza el nodo 1 porque la potencia de recepción del nodo 1 no tiene el alcance necesario.

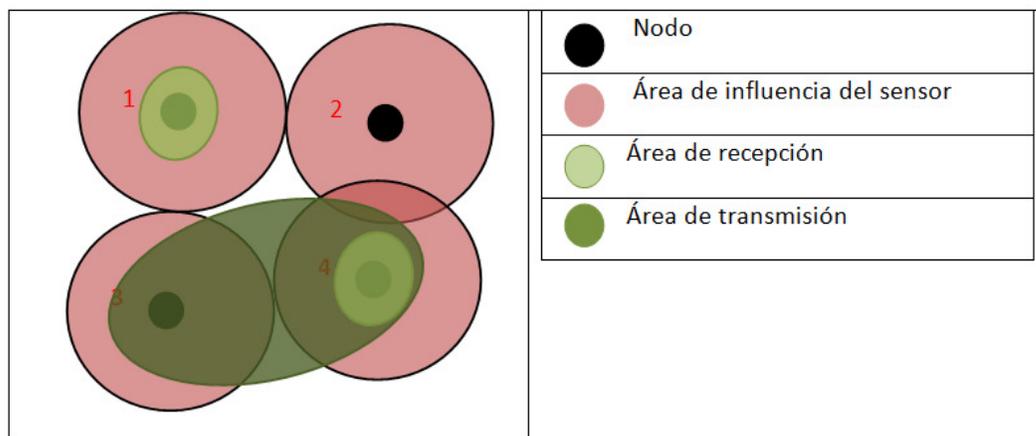


Figura 1.2: Nodos con sus respectivas áreas de influencia, potencia de transmisión y potencia de recepción

La tesis se divide en cuatro grandes bloques. El primero de ellos consiste en seleccionar de la totalidad de los nodos de la red solamente algunos que hagan la función de sensado y enrutamiento mientras los demás se encuentran en modo inactivo con el fin de ahorrar energía. El criterio para seleccionar los nodos es la distancia al centroide, la energía restante y el área cubierta. El segundo bloque es la generación de caminos disyuntos, esto con el fin de volver la red tolerante a fallas, para ello se genera una topología de árbol según el CTP [7] y posteriormente se crean los caminos disyuntos a través de las diferentes ramas con el algoritmo *Nto1* [8]. El tercer bloque crea matrices de comunicación que soporten potencias asimétricas, esto se realiza definiendo una métrica para el número de saltos desde los nodos hacia el sink. Finalmente se evalúa el sistema comparando el algoritmo desarrollado con CTP, y adicionalmente se genera un mapa de riesgo de pérdida de conectividad. En el capítulo 3 se explica con detalle cada uno de los bloques.

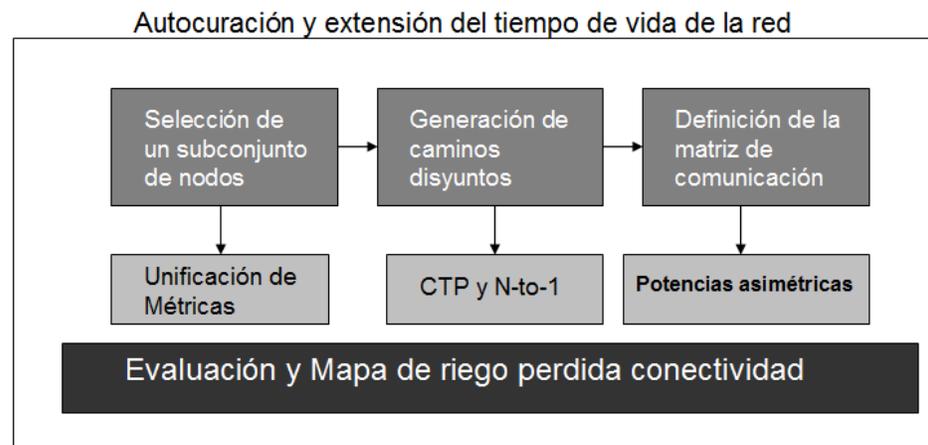


Figura 1.3: Diagrama de Bloques General

## Capítulo 2

# Marco Teórico

En este capítulo se muestra el estado del arte de las WSN. Específicamente se muestra en la sección 2.1 las diferentes técnicas de generación de topología, en la sección 2.2 se muestran los algoritmos más usados para la selección de un subconjunto de nodos, en la sección 2.3 se muestran algunos autores que han trabajado en la generación de caminos disjuntos, en la sección 2.4 se muestra la técnica usada para generar los mapas de riesgo de pérdida de conectividad y en la sección 2.5 se muestra cómo se define el vecindario de los nodos según las potencias asimétricas de transmisión y recepción.

### 2.1. Generación de Topología

En esta sección se explican los algoritmos más representativos de generación de topología, y luego de evaluar las posibilidades se selecciona el Collection Tree Protocol [7], el cual propone un algoritmo para la construcción y mantenimiento de un árbol.

Dado que la potencia de transmisión de los nodos es limitada, se requiere de técnicas que permitan transmitir la información entre puntos distantes en las WSN. Por lo tanto, los mensajes se ven obligados a ser transmitidos entre varios nodos para llegar desde la fuente de información hasta el destino. Contruir estos caminos teniendo en cuenta un gasto de energía distribuido entre los nodos de la red y garantizando un mínimo nivel de servicio es el objetivo principal del enrutamiento en las WSN.

Un hecho crítico en la construcción de topología es que si a un nodo se le agota la energía pierde la capacidad no solo de transmitir paquetes propios, sino de nodos vecinos, lo cual afecta el desempeño de toda la red. Otro hecho fundamental es que el canal inalámbrico presenta grandes tasas de error por bit, lo cual puede repercutir en el número de retransmisiones y por lo tanto puede aumentar dramáticamente el gasto de energía por nodo.

La entrega de los datos al sumidero pueden ser de tres formas distintas. La primera es basada en tiempo, lo cual significa que cada periodo de tiempo determinado los nodos van a reportar información al sumidero. La segunda es basada en eventos, esto quiere decir que cada vez que ocurra un evento dentro del área de interés los nodos van a reportar información al sumidero, un evento puede ser un incremento de la temperatura de 10 grados centígrados. Y la última forma de reportar datos es por petición del usuario, donde los nodos son informados a través de un mensaje que el usuario necesita información en ese momento.

### 2.1.1. Clasificación de Protocolos de Enrutamiento

Los protocolos de enrutamiento se clasifican en:

1. Enrutamiento Proactivo o Reactivo: Los protocolos proactivos encuentran rutas constantemente de la fuente de información al sumidero antes de que la ruta se necesite, es decir, si A desea transmitir un paquete a B, en los protocolos proactivos, en ese momento, ya existe una ruta disponible de A hacia B. Mientras que los protocolos reactivos únicamente buscan rutas de la fuente de información hacia el sumidero cuando ésta se necesita, es decir, trabajan por demanda. Si bien los protocolos reactivos generan un retardo en la entrega de paquetes mientras buscan una ruta disponible, los protocolos proactivos consumen una cantidad de energía considerable buscando rutas que posiblemente nunca se utilicen.
2. Enrutamiento jerárquico o plano: En el enrutamiento plano todos los nodos tienen funciones iguales y la distribución de la información se realiza a los nodos alcanzables. De otro lado el enrutamiento jerárquico se usa en redes que tienen miles de nodos donde se hace indispensable asignar funciones diferentes a los nodos tomando en cuenta su posición geográfica, cantidad de energía restante, cantidad de nodos vecinos, entre otros.

Existen cuatro categorías típicas para generalizar los protocolos de enrutamiento, estas son:

1. Enrutamiento Data-Centric: En este tipo de enrutamiento el sumidero envía mensajes pidiendo información a determinados nodos en diferentes áreas. La técnica SPIN es un ejemplo clásico de enrutamiento de tipo data-centric. En SPIN un nodo que tiene nueva información envía un mensaje informando su situación actual, posteriormente los nodos interesados en obtener la nueva información envían un mensaje haciendo el requerimiento. Lo interesante es que el nodo que posee la información envía una parte pequeña de ella, con el fin de que el receptor tenga criterio para definir si amplía o no la información recibida. Este algoritmo corrige problemas como tener información redundante y *overlapping*; también se agrega un módulo de manejo de energía para tomar decisiones basados en esta importante variable.

2. Enrutamiento Jerárquico: Se usa cuando el área de interés es muy amplia y se necesita que la red tenga escalabilidad, ya que de lo contrario se pueden tener problemas como altos tiempos de convergencia en los algoritmos, alto consumo de memoria, tiempos grandes de latencia e inestabilidad de la comunicación. El algoritmo más reconocido de este tipo es LEACH en el cuál los nodos se organizan en *clusters*, por lo tanto, existe un enrutamiento *intra-cluster* y otro *inter-cluster* que dirige los mensajes al sumidero. Es muy común que la cabeza del *cluster* agregue los datos con el fin de reducir la redundancia y el gasto innecesario de energía asociado.
3. Enrutamiento basado en localización: Este tipo de enrutamiento hace uso de GPS o algoritmos de localización de nodos para conocer la posición de cada uno dentro de la red. Al tener la posición de cada nodo se puede ajustar la potencia de transmisión para hacer un mejor uso de la energía y dirigir los paquetes a través de nodos específicos para alcanzar determinada zona de interés. Un algoritmo clásico de este tipo de enrutamiento es GEAR que tiene como métrica la cantidad de energía restante y la distancia a la zona donde se desea enviar el mensaje, de esta forma, si se tienen dos sumideros se puede tomar la mejor decisión energéticamente hablando al tener en cuenta la distancia.
4. Enrutamiento multicamino: Cuando se tienen aplicaciones que demandan gran cantidad de ancho de banda y tasas bajas de delay se recurre al enrutamiento multicamino. Dado que se tienen varios caminos de la fuente de información al sumidero el ancho de banda se incrementa y la red se vuelve tolerante a fallos. Los múltiples caminos pueden tener nodos disyuntos o links disyuntos, es decir, que no comparten nodos en el primer caso o que no comparten links en el segundo caso.

### 2.1.2. Collection Tree Protocol (CTP)

El Collection Tree Protocol [7] [9] es un algoritmo de generación de topología de tipo proactivo, cuyo objetivo es la construcción y el mantenimiento de un árbol a través del envío de paquetes informativos de enrutamiento llamados *beacons*. El algoritmo se divide en tres módulos con funciones específicas: *Routing Engine(RE)*, *Forwarding Engine(FE)* y *Link Estimator(LE)*. La implementación de este algoritmo para Castalia fue hecha por los autores de la referencia [7] donde el módulo global de enrutamiento *CtpNoe* encierra los submódulos RE, FE, LE, CTP y Dual Buffer. La función de los principales submódulos se explica a continuación:

1. *Link Estimator*: Su principal responsabilidad es la determinación de la calidad del link(ETX) de comunicación inbound y outbound entre un nodo y sus vecinos. La medida inbound se establece estadísticamente haciendo una división entre los paquetes efectivamente recibidos sobre el número de paquetes transmitidos por el vecino. De la misma forma, la medida outbound se establece haciendo una división entre los paquetes recibidos efectivamente por

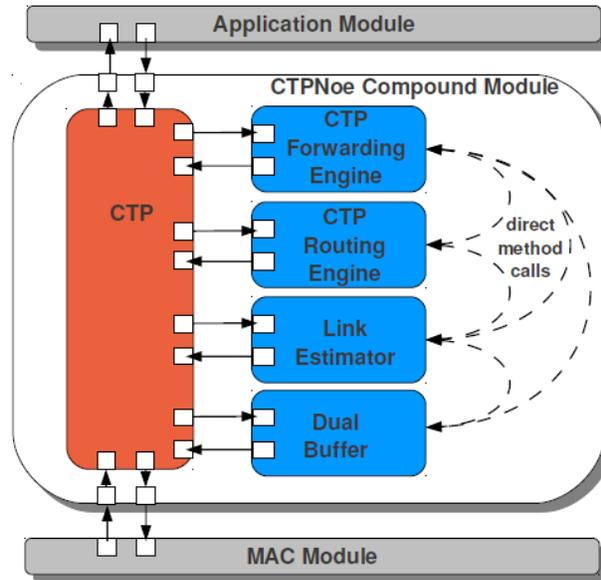


Figura 2.1: Módulos del Collection Tree Protocol. Figura tomada de [7].

el nodo vecino y el número de transmisiones. Esta información se guarda en una tabla que contiene el identificador del nodo vecino y el valor correspondiente de ETX. Los *beacons* tienen en su cabecera el número de secuencia *SeqNo* con el fin de poder calcular el ETX. Así, contando el número de paquetes recibidos de un nodo vecino y comparándolo con el número de secuencia se puede establecer el número de paquetes perdidos.

2. *Routing Engine*: La principal tarea del módulo es seleccionar un padre dentro del árbol hacia el cuál serán enviados los datos, esto lo hace a través del envío de *beacons*, llenando la tabla de enrutamiento y actualizándola. La frecuencia con la que se envían *beacons* se define a través del algoritmo Tricke, es decir, que el envío se reduce progresivamente. Una petición de descubrimiento de rutas hace que el algoritmo inicie nuevamente. Los *beacons* llevan información del ETX, son usados para llenar la tabla de enrutamiento y decidir quién va a ser el padre de cierto nodo. El proceso de selección de un padre se inicia cuando un *beacon* llega, un vecino es inalcanzable, un vecino deja de estar congestionado o el nodo no tiene rutas hacia el sink. Dentro de los vecinos de un nodo los posibles padres son los que no están congestionados y no son hijos de dicho nodo. Entre estos nodos se elige el que tiene menor valor de ETX.
3. *Forwarding Engine*: El objetivo de este módulo es enviar tanto los paquetes de datos que son

generados por el mismo nodo, como los paquetes de datos de otros nodos dentro de la red. Este módulo también se encarga de detectar y eliminar paquetes duplicados y loops. Para hacer el envío de datos se tiene una cola de tipo FIFO de 13 posiciones, donde el nodo espera un ACK antes de remover el paquete enviado de la cola, si no llega el ACK se procede a hacer la retransmisión. El número máximo de retransmisiones para un paquete es de 30. En el caso de que el número de retransmisiones sea permanentemente grande, la cola se va a ir llenando. Cuando la cola tiene más de la mitad de paquetes en espera se dice que el nodo está congestionado. Para detectar paquetes duplicados el nodo evalúa el *SeqNo* y el origen del paquete y lo compara con un cache. Para detectar un loop el nodo evalúa el ETX de sus hijos, si el ETX es menor quiere decir que existe un loop dentro de la red. Ante un loop el procedimiento es activar un timer de BackOff y pedir una actualización de topología.

## 2.2. Selección de un Subconjunto de Nodos

La selección de un subconjunto de nodos se realiza de la siguiente manera. Primero se agrupan los nodos en *clusters* por medio de la técnica *kmeans* [10], luego se selecciona la cabeza del *clusters* con PSO [11] aplicado a las métricas enunciadas en la sección 2.2.3.

### 2.2.1. Creación de Clusters con Kmeans

En una red de sensores inalámbricos cuyos nodos están distribuidos aleatoriamente en un campo de cientos de metros de largo y ancho es útil dividir el conjunto de nodos en subconjuntos más pequeños que tengan proximidad geográfica, con el fin de que las distancias de transmisión disminuyan y se contribuya al ahorro de energía. A esta técnica de dividir el conjunto de nodos en subconjuntos más pequeños con proximidad geográfica se le denomina *clustering* y ha sido estudiado ampliamente [10] debido a que las redes que lo aplican suelen incrementar su tiempo de vida y volverse más escalables.

La figura 2.2 muestra cuatro *clusters*, cada uno de diferente color, y un sumidero al cual es dirigida la información recolectada por todos los nodos de la red. La información llega al sumidero proveniente de las cabezas de los *clusters*(CH), estos últimos son los únicos nodos dentro de un *cluster* que se pueden comunicar directamente con el sumidero. Si otro nodo tiene información para transmitir ésta primero debe ser entregada al CH quien se encarga de transmitirla finalmente al sumidero. En el caso de que un CH no tenga la potencia suficiente para llegar directamente al sumidero se utiliza la técnica multisalto, es decir, que los CH se usan entre sí como retransmisores. Por ejemplo, el CH del *cluster* violeta no tiene la potencia suficiente para llegar al sumidero en un solo salto, por lo tanto, utiliza al CH del *cluster* amarillo como retransmisor.

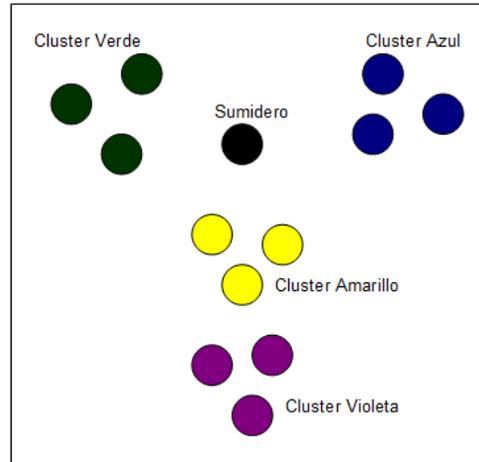


Figura 2.2: Creación teórica de clusters

La técnica *kmeans* es utilizada para la creación de *clusters* y consiste en los siguientes pasos. Primero se posicionan en el campo  $k$  centroides de manera aleatoria, luego se asigna cada nodo al centroide más cercano. Posteriormente se recalcula la posición de los centroides dentro de cada *cluster* tomando un promedio de las posiciones en  $x$  y en  $y$ . Si la posición de algún centroide cambia se debe volver a asignar los nodos a los centroides, de lo contrario el algoritmo finaliza quedando creados los *clusters*. La figura 2.3 muestra los pasos anteriormente descritos.

### 2.2.2. Particle Swarm Optimization (PSO)

El PSO y sus diferentes variantes han sido ampliamente usados en WSN [11], la siguiente tabla muestra algunas aplicaciones concretas.

Cuadro 2.1: Aplicaciones de PSO en WSN. Tabla tomada de [11].

Task of PSO	Optimization Criterion	Algorithm	Reference
Position Stationary Nodes	Max. Coverage	PSO-Voronoi	[12]
Localize Nodes	Min. Localization Error	PSO-Loc	[13]
Elect Cluster-Heads	Min. intra-cluster distance	PSO-Clustering	[14]
Allocate Optimal Transmission Power	Min. Energy expenditure and error prob.	PSO-Opt-Alloc	[15]]

El PSO es un método de optimización inspirado en la naturaleza que emula el comportamiento

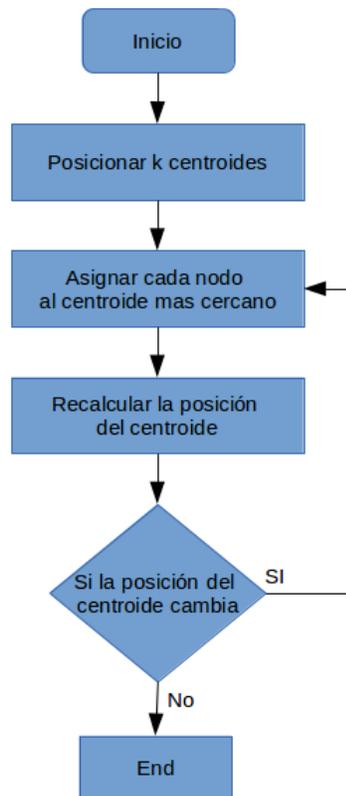


Figura 2.3: Algoritmo kmeans

de parvada de aves o un cardumen de peces. El PSO fue planteado por primera vez por Kennedy y Eberhart [16] en 1995. El PSO inicia con un conjunto de soluciones arbitrario llamado la población inicial la cual se posiciona aleatoriamente en el hiperespacio del problema y cuyo movimiento está determinado por la ecuación de velocidad y de posición que se verán más adelante. Una solución óptima al problema se busca actualizando la posición de cada una de las partículas.

Comparativamente el PSO tiene menos parámetros que los algoritmos GAs (Genetic Algorithm), no usa operadores evolutivos como mutación y crossover, y adicionalmente no requiere codificación y decodificación de números binario por lo cual resulta más fácil implementarlos en el computador [17].

EL PSO define los siguientes conceptos. Una partícula es un individuo de la manada que tiene una posición dentro del hiperespacio, cada una representa un punto de diseño que es una posible solución del problema. La posición de la partícula se da en coordenadas y su velocidad es la tasa

a la cual se mueve. Y el *Swarm Leader* es la partícula con la mejor posición dentro de la manada. El movimiento de las partículas se define según las ecuaciones (2.1) (2.2)

$$v_i(k+1) = \phi(k)v_i(k) + \alpha_1[\gamma_{1i}(p_i - x_i(k))] + \alpha_2[\gamma_{2i}(G - x_i(k))] \quad (2.1)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (2.2)$$

Cuadro 2.2: Parámetros del Algoritmo PSO

Variable	Descripción
$i$	Índice de la partícula
$k$	Índice de tiempo discreto
$x$	Posición de la partícula $i$
$v$	Velocidad de la partícula $i$
$p$	Mejor posición de la partícula $i$
$G$	Mejor posición entre todas las partículas
$\gamma_{1,2i}$	Número aleatorio entre 0 y 1
$\alpha_{1,2}$	Constante de aceleración
$\phi$	Función de inercia

### 2.2.3. Métricas para la Selección de un Subconjunto de Nodos

Con el fin de seleccionar un subconjunto de nodos que extiendan el tiempo de vida de la red manteniendo la cobertura se han estudiado 4 autores.

El primer autor es Xu Wang *et al* [18] quien realiza métricas de consumo de energía, efectividad de cobertura y tiempo de vida de la red. La métrica de la energía de transmisión que plantea el autor es la ecuación (2.3).

$$e_i^T = p_i^T t_0^T = \frac{p_0^T (4\pi)^2 \beta l_i^2 t_0^T}{G_f G_r \lambda^2 l_0^2} \quad (2.3)$$

Las variables de la ecuación (2.3) se explican en la tabla 2.3. La energía total del sistema se muestra en la ecuación (2.4).

$$\phi_E = \sum_{i=1}^N e_i s_i \quad (2.4)$$

Cuadro 2.3: Parámetros de la Métrica de la Energía de Transmisión

Variable	Descripción
$l_i$	Distancia entre nodos
$G_r$	Ganancia de recepción
$G_t$	Ganancia de transmisión
$\lambda$	Longitud de onda
$p_0^T$	Potencia de transmisión
$\beta$	Atenuación del sistema
$t_0^T$	Tiempo de la comunicación

Donde  $s_i = 0$  si el nodo está en modo sleep y  $s_i = 1$  si el sensor está activo. El área de cobertura tiene como métrica la ecuación (2.5).

$$\varphi_c = \frac{N_{effective}}{N_{all}} \quad (2.5)$$

Donde  $N_{all}$  es la cantidad total de grillas en las que se divide el área de cobertura y  $N_{effective}$  es el número de grillas cubiertas efectivamente. Ahora, se hace la relación entre el porcentaje de cobertura y la energía gastada por el sistema, siendo ésta la función objetivo a maximizar para prolongar el tiempo de vida de la red.

$$\phi = \frac{\varphi_c}{\phi_E} \quad (2.6)$$

Esta es la métrica se usa para establecer el compromiso entre porcentaje de cobertura y cantidad de energía utilizada en la comunicación.

El segundo autor Shih *et al* [19] hace la selección del subconjunto de nodos basándose en los diagramas de Voronoi y en la energía restante de cada nodo, a la suma de estos dos criterios les da el nombre de DASS (Distributed Active Sensor Selection Scheme). DASS es un esquema de selección de subconjunto de nodos que elige un nodo inicializador al cual llama *init* y posteriormente encuentra los vecinos de compañía (Companion Neighbors) a la vez que crea el diagrama de Voronoi. El esquema DASS garantiza cobertura completa.

El nodo inicializador *init* encuentra una bisectriz perpendicular  $b_{si}$  entre el nodo *init* y cada uno de sus vecinos como lo muestra la siguiente figura. Posteriormente garantiza que los puntos de intersección de cada  $b_{si}$  estén dentro del rango de sensado (circunferencia alrededor de  $s$ ). Luego se calcula el segmento de cobertura perimetral  $|\Gamma_j|$  para cada  $b_{si}$ , y se garantiza que se elijan los más

pequeños ya que esto permite que la celda de Voronoi  $VC(s)$  sea la más grande. Adicionalmente se verifica que los  $[\alpha_{l_j}, \alpha_{r_j}]$  elegidos cubran el perímetro del nodo de  $[0 \cdot 2\pi]$ . La energía restante  $E_{r_j}$  de cada nodo se incluye dentro del peso  $w_j$  que está definido como:

$$w_j = \lambda \frac{\text{abs}(|\Gamma_j| - \frac{\pi}{3})}{\frac{2\pi}{3}} - (1 - \lambda)(1 - E_{r_j}) \quad (2.7)$$

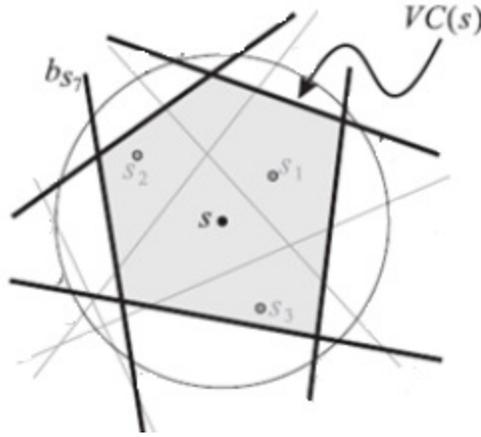


Figura 2.4: Creación de diagramas de voronoi con el esquema DASS. Figura tomada de [19].

El tercer autor es Pedro Wightman *et al* [20] quienes nos presentan un algoritmo denominado A3 con el cual se llega a tener una topología de árbol que extiende el tiempo de vida de la red. El algoritmo A3 selecciona un nodo inicializador o nodo padre que descubre a los vecinos y les envía mensajes de apagado para ponerlos en modo sleep según el criterio definido en la ecuación (2.8)

$$M_{x,y} = W_E \frac{E_x}{E_{max}} + W_D \frac{RSSI_y}{RSSI^*} \quad (2.8)$$

Donde  $W_E$  es el peso de la energía restante del nodo  $E_x$ ,  $E_{max}$  es la energía máxima inicial del nodo,  $W_D$  es el peso de la distancia al nodo padre,  $RSSI_y$  es la intensidad de señal recibida del nodo padre y  $RSSI^*$  es la mínima intensidad de señal para asegurar conectividad. Los pesos  $W_E$  y  $W_D$  se eligen de acuerdo al criterio que se le quiera asignar mayor importancia.

El cuarto autor A. Latiff *et al* [21] presenta un enfoque similar al anterior dado que presenta una métrica basada en energía y distancia para la selección de nodos, sin embargo no usa el algoritmo A3 sino el PSO. Latiff propone una topología basada en *clusters* donde el *cluster head* es el encargado de hacer la agregación de datos para disminuir la energía invertida en transmisión.

La selección del *cluster head* se hace con la métrica definida en la ecuación (2.9).

$$cost = \beta f_1 + (1 - \beta) f_2 \quad (2.9)$$

$$f_1 = \max_{k=1,2,\dots,K} \left\{ \sum_{n_i \in C_{p,k}} \frac{d(n_i, CH_{p,k})}{|C_{p,k}|} \right\} \quad (2.10)$$

$$f_2 = \frac{\sum_{i=1}^N E(n_i)}{\sum_{i=1}^N E(CH_{p,k})} \quad (2.11)$$

Donde  $f_1$  es la distancia máxima promedio euclidiana de los nodos a su *cluster head* asociado,  $|C_{p,k}|$  es el número de nodos que pertenecen al *cluster*  $C_k$ ,  $f_2$  es la función de la relación entre la energía de todos los nodos en la red con la energía de los *cluster heads* actuales. De la misma manera que el autor anterior  $\beta$  se elige de acuerdo al criterio que se le quiera asignar mayor importancia.

## 2.3. Generación de Caminos Disyuntos

A continuación se presentan algunos autores que han trabajado el tema de caminos disyuntos. La técnica que se va a implementar dentro de la tesis es *N-to-1 Multipath Routing* la cual hace uso de un árbol para generar los caminos disyuntos. Los árboles son creados con la técnica *Collection Tree Protocol* explicada en la sección 2.1.2.

### 2.3.1. Algunos Autores Representativos

Dentro de los autores representativos se destaca M. Maimour [22]. Este autor estudia las Redes de Sensores Inalámbricas Multimedia (WMSN) por lo cual le da una importancia especial al ancho de banda. Para incrementar el ancho de banda se propone crear múltiples caminos disyuntos que tengan la característica de no interferirse entre sí. Se parte de un único camino y solamente son creados los caminos disyuntos cuando sea necesario, es decir, cuando exista congestión o el ancho de banda se quede corto. Al seleccionar un camino los nodos que generen interferencia son puestos en el estado sleep con el fin de ahorrar energía. El método para encontrar caminos disyuntos es MR2 (Maximally Radio-Disjoint Multipath Routing) donde el sumidero hace un broadcast para encontrar el camino primario hacia la fuente, durante este recorrido cada nodo actualiza su tabla de enrutamiento haciendo una entrada para el sumidero. Posteriormente la fuente interroga a sus vecinos por un camino hacia el sumidero con menor métrica o costo, la métrica es de libre escogencia. Si se desea un segundo camino se repite el proceso pero no se tiene en cuenta a los

nodos que generan interferencia.

Cuando se utilizan caminos disyuntos puede generarse ineficiencia energética debido a que unos caminos son más largos, por lo tanto D. Ganesan *et al* [23] propone que los caminos sean no completamente disyuntos, es decir, que se generen caminos trenzados donde el camino principal y los auxiliares pueden compartir algunos nodos, figura 2.5. Para construir los caminos trenzados se supone que cierto nodo  $n(k)$  del camino principal va a fallar, por lo tanto es necesario encontrar un camino auxiliar que no incluya dicho nodo pero que no sea completamente disyunto del primero, como es de esperar este camino va a estar cerca geográficamente del camino primario y por lo tanto va a gastar una cantidad similar de energía. En la figura 2.5 resulta que  $n(k)$  es reemplazado por su vecino  $a(k)$  en caso de falla creando el camino auxiliar conformado por los nodos de color verde.

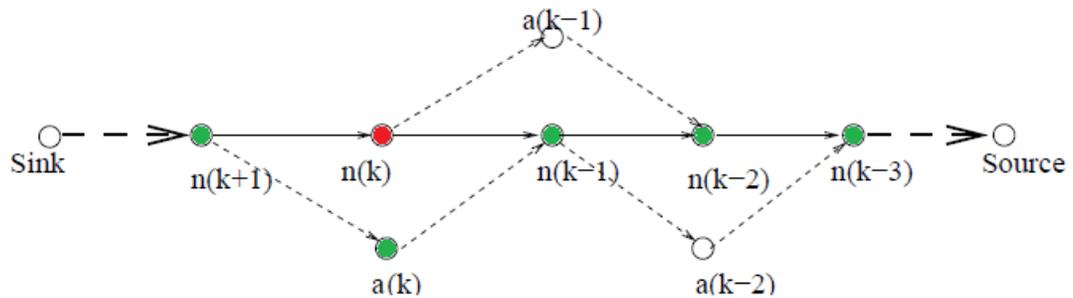


Figura 2.5: Caminos Trenzados para Soportar la Falla de Algunos Nodos. Tomado de [18].

De otro lado X. Hou *et al* [24] retoma la técnica de la inundación y propone un método para encontrar caminos disyuntos llamado Enrutamiento Multicamino Basado en Etiquetas (Label-Based Multipath Routing LMR). En LMR el sumidero difunde un mensaje de broadcast que contiene una etiqueta, cada nodo por donde pasa dicho mensaje guarda en su tabla de enrutamiento la etiqueta que corresponde a un identificador del camino. Ya teniendo una idea de los posibles caminos para llegar a la fuente el sumidero introduce un mensaje de refuerzo a un camino preferido.

### 2.3.2. N-to-1 Multipath Routing

La técnica que se va a usar dentro de la tesis es *N-to-1 Multipath Routing* [8] la cual hace uso de un árbol para generar los caminos disyuntos. Los árboles son creados con la técnica *Collection Tree Protocol* explicada en la sección 2.1.2. La forma más primitiva de crear un árbol es por el mecanismo de inundación, el cual consiste en que el sumidero transmite un mensaje de broadcast,

este mensaje va saltando de nodo en nodo. Lo interesante es que el nodo que recibe el mensaje decide que su padre es el nodo vecino que se lo transmitió. Sin embargo, el árbol que resulta tiene el problema de que no es tolerable a fallos, ya que la muerte de un nodo padre perjudica a todos los hijos asociados. Con el fin de hacer el árbol tolerable a fallas se incluyen caminos disyuntos dentro del mismo a través de la técnica *N-to-1 Multipath Routing*

La técnica *N-to-1 Multipath Routing* se compone de dos fases las cuales se explican a continuación:

1. *Branch Aware Flooding*: La forma general de un mensaje es  $mtype, mid, nid, bid, cst, path$ . La variable  $mtype$  identifica si el camino es primario(RPRI) o secundario, en esta fase del algoritmo los caminos siempre son primarios. La variable  $mid$  es el número de secuencia del mensaje. La variable  $nid$  es el identificador del nodo que envía el mensaje. La variable  $bid$  es el identificador de la rama, las ramas se identifican por el nodo que pertenece a esa rama que está más cerca del sumidero. La variable  $path$  contiene la secuencia de nodos por los cuales el mensaje a viajado. Y por último la variable  $cst$  es el costo del camino.

El sumidero inicia la actualización de las rutas enviando el siguiente mensaje de broadcast  $RPRI, mid, Sink, 0, 0, (Sink)$ , el cual significa que es una ruta primaria que conduce hacia el sumidero. Los nodos que escuchan este mensaje, por ejemplo  $z$ , marcan el sumidero como su padre y aprenden la ruta inversa  $path$ . Luego envían el siguiente mensaje  $RPRI, mid, z, z, cst + cst(z, parent(z)), path + z$  para seguir informando al resto de los nodos el camino correcto. Este proceso se repite.

Se tiene el concepto de nodos primos, los cuales son nodos vecinos en ramas distintas. Si llega un mensaje de actualización de un nodo primo que tiene un costo menor al camino primario, este camino se incluye dentro del conjunto de caminos disyuntos. De esta forma se pueden tener caminos disyuntos como los que muestra la siguiente imagen. La imagen tiene un sumidero de color amarillo y tres ramas de color negro, azul y verde, las líneas rojas representan caminos disyuntos a través de nodos primos.

2. *Multipath Extension Flooding*: La primera etapa del algoritmo encuentra caminos disyuntos únicamente a través de los nodos primos. Esta etapa encuentra más caminos disyuntos por medio de un segundo tipo de mensaje, RALT, que significa camino alternativo. Cuando un nodo escucha un camino alternativo verifica si éste proviene de su padre, si es así el mensaje se ignora. También se descarta el mensaje si el nodo mismo ya está incluido en el camino hacia el sumidero. Si las condiciones anteriores son falsas el nodo agrega la nueva ruta, el cuál es un camino disyunto que no es propio de un primo. Por ejemplo, en la figura anterior el nodo  $w$  publica un camino alternativo  $w - v - q - k - e - a - sink$ , el cual es escuchado por  $r$

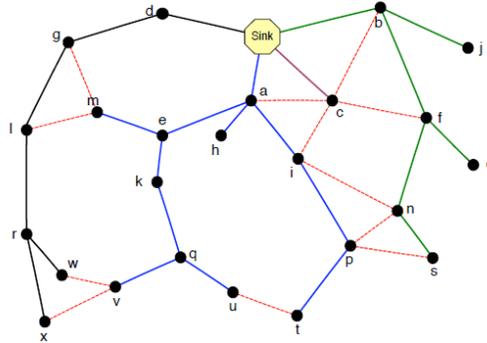


Figura 2.6: Árbol con Caminos Disyuntos. Tomado de [8]. .

generándose el nuevo camino  $r - w - v - q - k - e - a - sink$ . Como se observa  $r$  y  $w$  no son primos, sin embargo, se pudo establecer un nuevo caminos disyunto. En esta segunda fase se encuentran nuevos caminos pero se agrega más tráfico de control a la red.

## 2.4. Generación de Mapa de Pérdida de Riesgo de Conectividad

Con el fin de poder tomar acciones preventivas se genera una métrica de pérdida de riesgo de conectividad que incluye las siguientes variables: Cantidad de nodos muertos, cantidad de nodos activos y cantidad de nodos en estado inactivo. El objetivo es tomar el área de interés y dividirla en cuadrados, se está trabajando con un área de  $60 \times 60$  metros la cual se divide en 6 cuadrados de  $10 \times 10$  metros. Posteriormente se hace un conteo en cada cuadrado de  $10 \times 10$  metros de la cantidad de nodos muertos, activos y en estado inactivo. La figura 2.7 muestra una distribución aleatoria de nodos, los nodos muertos son los de color rojo, los activos los de color verde y los que están en estado sleep son los de color amarillo.

La métrica específica se define en la sección 5.2. Luego de aplicada la métrica cada cuadrado va a tener asignado un valor y se hace una interpolación con la técnica de Kriging.

La técnica de Kriging es un método estadístico de interpolación cuyo objetivo es predecir los valores de la variable de interés en lugares no muestreados. Se caracteriza por ser un predictor lineal, insesgado y de varianza mínima. Se ajusta al problema que se tiene ya que genera una superficie estimada a partir de un conjunto de puntos dispersados. Kriging presupone que la distancia entre los puntos de muestras reflejan una correlación espacial que puede utilizarse para explicar la variación de la superficie. También es útil Kriging para el problema específico de esta

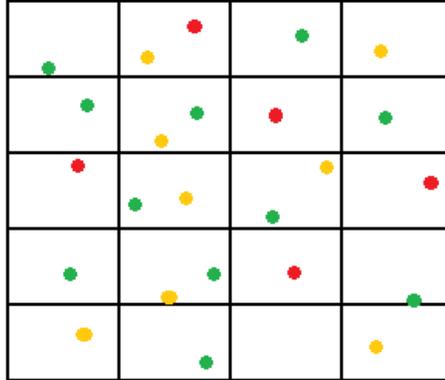


Figura 2.7: Conteo de nodos muertos, activos y en estado sleep

tesis porque ajusta una función matemática a todos los puntos dentro de un radio específico para determinar el valor de salida para cada ubicación. La fórmula de Kriging se define en la ecuación (2.8).

$$\hat{Z}(S_0) = \sum_{i=1}^N \lambda_i Z(S_i) \quad (2.12)$$

donde  $Z(S_i)$  es el valor medido de la ubicación  $i$ ,  $\lambda_i$  es un peso para el valor correspondiente  $Z(S_i)$ ,  $S_0$  es la ubicación de la predicción y  $N$  es la cantidad de valores medidos. Luego de aplicada la técnica de Kriging se debe obtener una superficie que muestre el riesgo de pérdida de conectividad, figura 2.8. De igual manera, el color verde indica que no hay riesgo de pérdida de conectividad y el color rojo indica un riesgo inminente.

## 2.5. Matrices de comunicación asimétricas

Castalia es el simulador que se va a utilizar durante el desarrollo de la tesis. Castalia permite elegir potencias de transmisión y de recepción asimétricas, por defecto el radio CC2420 trae como potencia de recepción -95dBm y la potencia de transmisión se puede variar según el cuadro 2.4.

Así mismo estas potencias de transmisión definen una distancia máxima de transmisión como lo muestra la figura 2.9

Durante el análisis de resultados en el capítulo 5 se varía la potencia de transmisión con el fin de ver los efectos generados en la red.

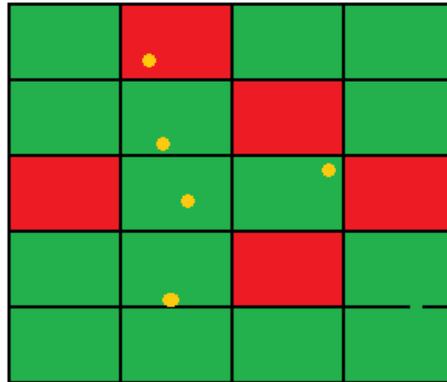


Figura 2.8: Interpolación de Kriging

Cuadro 2.4: Potencia de TX y consumo

Potencia de TX(dBm)	Consumo (mW)
0	57,42
-1	55,18
-3	50,69
-5	46,2
-7	42,24
-10	36,3
-15	32,67
-25	29,04

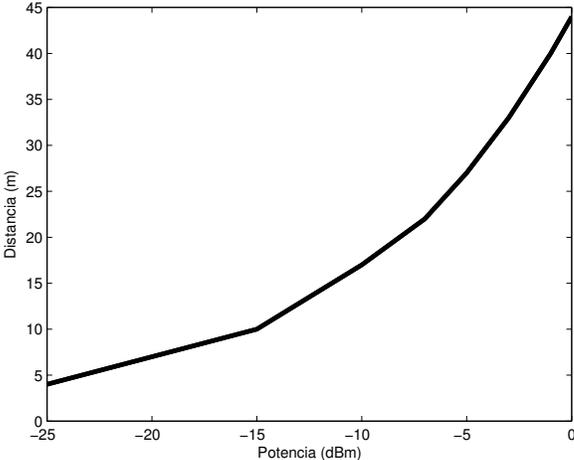


Figura 2.9: Distancia máxima de transmisión a diferentes potencias

## Capítulo 3

# Especificaciones

En esta sección se expone un diagrama de bloques para DACA (Disjoint path And Clustering Algorithm), la técnica desarrollada durante la tesis. Adicionalmente se muestra un diagrama de bloques del simulador Castalia, el cual es utilizado para la implementación y evaluación de DACA. Este diagrama de bloques muestra cómo se implementó DACA en Castalia utilizando el modelo de capas que propone éste último. También se muestran las variables de entrada y salida que produce cada uno de los módulos de DACA.

### 3.1. Diagrama de Bloques DACA

Esta sección está dedicada a presentar el diagrama de bloques de la técnica desarrollada durante la tesis cuyo nombre es DACA. La figura 3.1 muestra el diagrama de bloques de DACA.

El primer bloque del diagrama de la figura 3.1 es el **árbol de comunicación inicial**. Este bloque, a través de la técnica CTP, permite construir un árbol de comunicación inicial que cuente con todos los nodos de la red. La utilidad del árbol es enviar información al sink como la posición y la energía restante de cada uno de los nodos para volver el algoritmo DACA centralizado. Se ha usado el algoritmo CTP para la construcción de este árbol inicial debido a que la métrica para la selección del padre es la calidad del link de comunicación, y dado a que se desea una red confiable ésta métrica nos permite alcanzar cierto nivel de confiabilidad.

Luego de que se cuenta con un árbol de comunicación inicial, el algoritmo DACA selecciona un subconjunto de nodos a través del trabajo mancomunado de dos bloques. La selección de un subconjunto de nodos es útil porque permite poner nodos en estado inactivo (*sleep*) con el fin de tener únicamente unos pocos nodos activos y de este modo ahorrar energía. El primero de estos bloques es la **creación de clusters** por medio del algoritmo Kmean. Kmeans es un algoritmo de cuatro pasos que divide los nodos en la cantidad de clusters que se desee, el objetivo de esto es poder activar únicamente un CH por cluster, lo cual se traduce en ahorro de energía. Kmeans

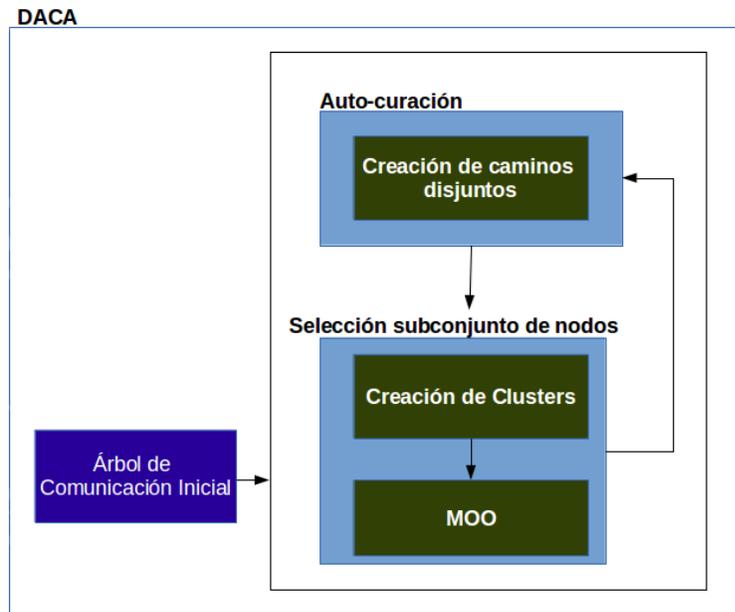


Figura 3.1: Diagrama de bloques de la técnica DACA

distribuye los clusters uniformemente en la red, lo cual garantiza que la cobertura se maximice. El segundo de estos bloques es la técnica multi-objetivo (**MOO**) PSO con la cual se selecciona un CH por cluster garantizando que se maximice la energía restante del nodo, la cobertura de la red y el inverso de la distancia a los demás nodos del cluster.

En este punto se cuenta con la red dividida en clusters y un CH seleccionado por cluster. Únicamente con este subconjunto de nodos (CHs) se crea un nuevo árbol haciendo uso de CTP nuevamente, mientras que los no-CH se ponen en estado inactivo. Sobre este árbol se corre el algoritmo N-to-1 para **crear caminos disjuntos**, y con ello la red se vuelve tolerante a fallos.

Cuando no se encuentran suficientes caminos disjuntos en un área específica es porque la densidad de la red en ese punto es baja, es decir, la cantidad de nodos por metro cuadrado es baja. Si se llega a presentar un caso donde no se encuentran suficientes caminos disjuntos es necesario que exista una realimentación entre el bloque de **creación de caminos disjuntos** y el bloque de **selección de subconjunto de nodos**, donde el primero le informe al segundo la situación de insuficiencia de caminos disjuntos, esto con el fin de que el segundo pueda seleccionar más de un CH por cluster para aumentar la densidad de la red.

Finalmente, el algoritmo DACA se evalúa a través de cuatro métricas que se definen en la sección 5.2, sin embargo, se adelanta que las métricas son: número de saltos hacia el sink, tiempo

de vida de la red, porcentaje de cobertura y riesgo de pérdida de conectividad.

### 3.2. Diagrama de Bloques Castalia

En la figura 3.2 se muestra la estructura del nodo en Castalia, la cual está organizada por capas. Se cuenta con la capa de aplicación, capa de enrutamiento, capa de enlace de datos y capa física. Adicionalmente se cuenta con el módulo de *physical Process* en el cual se simula la generación de información de la red, se tiene el módulo de *Resource Manager* en el cual se administra el gasto de energía, se cuenta con el módulo de *Mobility* en el cual se maneja el movimiento de los sensores y existe el módulo del *wireless channel* en el cual se simula las condiciones del canal inalámbrico.

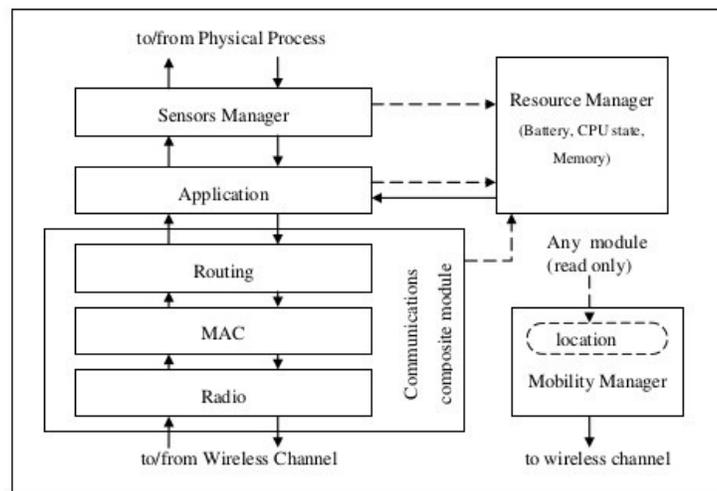


Figura 3.2: Estructura del nodo en Castalia

La figura 3.3 muestra la implementación de DACA en Castalia. En la capa de aplicación del sink se implementaron los algoritmos PSO y Kmeans con el fin de que éste último pudiera hacer la selección del subconjunto de nodos de forma centralizada, es decir, pudiera crear los clusters y elegir los CHs de toda la red.

En la capa de comunicación, específicamente en el submódulo de enrutamiento, se usó la implementación de CTP de los autores [7] para construir y crear un árbol con raíz en el sink. Adicionalmente, en este mismo submódulo se implementó sobre CTP el algoritmo N-to-1 para que los nodos pudieran encontrar caminos disjuntos. Los módulos del *physical process*, *sensor manager* y *mobility manager* no se usaron debido a que la tesis no tiene como objetivo la evaluación del sensor

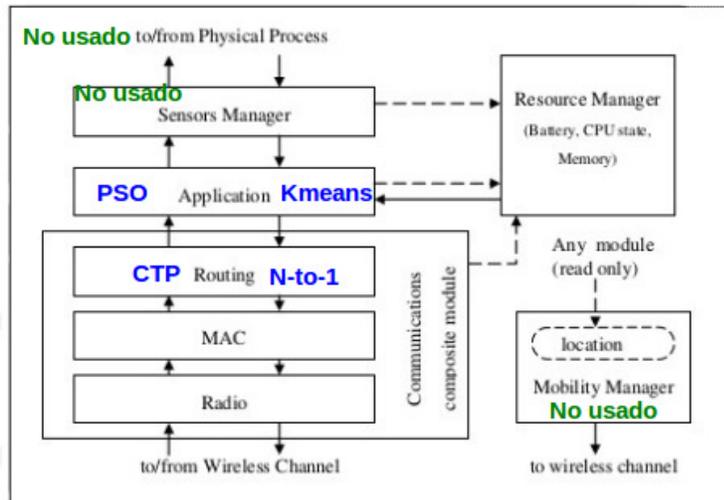


Figura 3.3: Implementación de DACA en Castalia

y tampoco considera nodos que tengan movimiento. Los módulos de mac, radio y *resource manager* son usados en la red, pero las modificaciones en el código son mínimas, dado que únicamente se leen variables como la energía restante, la potencia empleada en la transmisión y recepción de paquetes, entre otros.

### 3.3. Especificaciones de DACA por módulo

En esta sección se especifican las variables de entrada y de salida propias de cada módulo de DACA. Iniciamos con el módulo árbol de comunicación inicial en la figura 3.4, en la cual se observa que las variables de entrada son el número de nodos, la potencia de transmisión y la potencia de recepción. CTP para la construcción del árbol inicial necesita saber cuántos nodos tiene la red y cuáles son las potencias que maneja el radio, mientras que la salida del módulo es una tabla de enrutamiento que indica cuál es el padre de cada nodo y cuál es el costo hacia ese padre, el costo en CTP es el ETX.

La figura 3.5 muestra las variables de entrada y salida para el módulo MOO. Este módulo implementa la técnica de optimización multi-objetivo PSO, la cual necesita como parámetros de entrada la energía inicial y restante de cada nodo, la posición de cada nodo para calcular la distancia a los diferentes centroides y la cobertura que proporciona cada uno de los nodos. Las variables de salida del PSO son la selección de un subconjunto de nodos, es decir, la selección de los CH y la selección de los nodos inactivos.



Figura 3.4: Variables de entrada y salida del árbol de comunicación inicial



Figura 3.5: Variables de entrada y salida del MOO

La figura 3.6 muestra las variables de entrada y salida para el módulo de creación de clusters. Este módulo implementa Kmeans, el cual necesita la información de la posición de cada uno de los nodos pero centralizada en el sink y el número  $k$  que es el número de clusters en la red. Como variable de salida este módulo indica cuál nodo pertenece a cada cluster.



Figura 3.6: Variables de entrada y salida de la creación de clusters

La figura 3.7 muestra las variables de entrada y salida para el módulo de creación de caminos disjuntos. Este módulo implementa el algoritmo N-to-1 y necesita como variable de entrada un árbol que únicamente utilice los CH, mientras que como variables de salida este algoritmo identifica a las ramas con un ID y adicionalmente encuentra caminos disjuntos para los nodos.

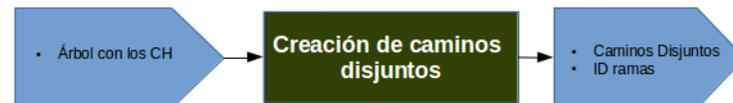


Figura 3.7: Variables de entrada y salida de la creación de caminos disjuntos

## Capítulo 4

# Desarrollo Teórico

Esta sección presenta la técnica desarrollada durante la tesis: *DACA, Disjoint paths And Cluster Algorithm*. DACA es un algoritmo de tres pasos cuyo objetivo es incrementar el tiempo de vida de la red manteniendo la cobertura, esto mediante el empleo de técnicas de auto-curación y el control de la topología en dos fases. Se resumen los tres pasos de DACA a continuación y posteriormente se explican al detalle.

1. **Construcción de un árbol de comunicación inicial:** Con la ayuda del algoritmo CTP [7] se construye un árbol con todos los nodos de la red el cual va a tener como raíz el sink. Éste árbol es útil porque a través de él se va a enviar la información de la energía restante y la posición de nodo con el fin de que el algoritmo DACA se vuelva centralizado. Esta información se envía a través de los paquetes de datos del CTP, por lo tanto va a llegar a la capa de aplicación del sink.
2. **Creación de cluster:** En este momento el sink tiene la información de la posición y la energía restante de todos los nodos, por lo tanto, ejecuta Kmeans [10] de forma centralizada. Con la ayuda de Kmeans se asigna cada uno de los nodos a un cluster, esto con el fin de seleccionar un CH por cluster y los no-CH ponerlos en modo inactivo para ahorrar energía e incrementar el tiempo de vida de la red. Se asume que el CH es capaz de caracterizar todo el cluster, por lo tanto, apagar los no-CH no implica una pérdida considerable de cobertura.
3. **Optimización multi-objetivo (MOO):** La optimización multi-objetivo realizada con el algoritmo PSO [25] permite seleccionar los CH maximizando la energía restante que tiene el CH, la cobertura que provee el CH y minimizando la distancia hacia los no-CH. Con lo anterior se garantiza que siempre se elija como CH el nodo que tenga más energía disponible, el nodo que permita mayor cobertura, es decir, el que menos sobrelape su radio de sensado con otros CH y el que esté más cerca del centroide, esto con el fin de reducir las distancias de transmisión en el caso de que en el futuro se piense implementar agregación de datos por el CH.

4. **Creación de caminos disjuntos:** Con los CH encontrados en el punto 3 la técnica CTP construye un nuevo árbol que cuente con la ventaja de utilizar únicamente un subconjunto de nodos, esto con el fin de ahorrar energía ya que la mayoría de nodos van a estar en modo inactivo. Con este nuevo árbol se crean caminos disjuntos con la técnica N-to-1 [8]. N-to-1 identifica las ramas del árbol con un ID y posteriormente cada nodo envía mensajes de broadcast con la intención de encontrar una rama diferente a la cuál él pertenece, si dicha rama es encontrada se dice que dicho nodo cuenta con un camino disjunto. Si algún nodo de la rama principal falla, los nodos hijos pueden acudir a los caminos disjuntos para enviar los datos al sink, con esto la red se vuelve tolerante a fallos e incrementa su tiempo de vida, el costo de ello es intercambiar algunos mensajes de más en la red.

#### 4.1. Construcción de un árbol de comunicación inicial

El primer paso en DACA es crear un árbol con raíz en el sink, el cuál utilice todos los nodos de la red. La función de este árbol inicial es permitir que la información de la energía restante y posición de cada nodo esté disponible en el sink, esto con el fin de que éste último ejecute Kmeans y PSO de manera centralizada, como se explica en las secciones 4.2 y 4.3 respectivamente.

El CTP [7] es el algoritmo utilizado para hacer la recolección de datos debido a que es un servicio confiable que permite reportar los datos a uno o más sinks. El objetivo principal del CTP es la construcción y el mantenimiento de un árbol para crear caminos con el fin de que los paquetes de datos lleguen al sink.

La construcción de un árbol está basado en la selección de un padre por nodo, y dicha selección está basada en una métrica, que en el caso del CTP puede ser la cantidad de saltos hacia el sink o la calidad del link de comunicación hacia el padre. Debido a que se desea una red confiable, se ha elegido en el desarrollo de la tesis, la calidad del link de comunicación hacia el padre como métrica del CTP. El CTP se divide en tres módulos los cuales se explican a continuación.

1. *Link Estimator:* Su principal responsabilidad es la determinación de la calidad del link (ETX) de comunicación inbound y outbound entre un nodo y sus vecinos. La medida inbound se establece estadísticamente haciendo una división entre los paquetes efectivamente recibidos sobre el número de paquetes transmitidos por el vecino. De la misma forma, la medida outbound se establece haciendo una división entre los paquetes recibidos efectivamente por el nodo vecino y el número de transmisiones. Esta información se guarda en una tabla que contiene el identificador del nodo vecino y el valor correspondiente de ETX. Los *beacons* tienen en su cabecera el número de secuencia *SeqNo* con el fin de poder calcular el ETX. Así, contando el número de paquetes recibidos de un nodo vecino y comparándolo con el número de secuencia se puede establecer el número de paquetes perdidos.

2. *Routing Engine*: La principal tarea del módulo es seleccionar un padre dentro del árbol hacia el cuál serán enviados los datos, esto lo hace a través del envío de *beacons*, llenando la tabla de enrutamiento y actualizándola. La frecuencia con la que se envían *beacons* se define a través del algoritmo Tricke, es decir, que el envío se reduce progresivamente. Una petición de descubrimiento de rutas hace que el algoritmo inicie nuevamente. Los *beacons* llevan información del ETX, son usados para llenar la tabla de enrutamiento y decidir quién va a ser el padre de cierto nodo. El proceso de selección de un padre se inicia cuando un *beacon* llega, un vecino es inalcanzable, un vecino deja de estar congestionado o el nodo no tiene rutas hacia el sink. Dentro de los vecinos de un nodo los posibles padres son los que no están congestionados y no son hijos de dicho nodo. Entre estos nodos se elige el que tiene menor valor de ETX.
  
3. *Forwarding Engine*: El objetivo de este módulo es enviar tanto los paquetes de datos que son generados por el mismo nodo, como los paquetes de datos de otros nodos dentro de la red. Este módulo también se encarga de detectar y eliminar paquetes duplicados y loops. Para hacer el envío de datos se tiene una cola de tipo FIFO de 13 posiciones, donde el nodo espera un ACK antes de remover el paquete enviado de la cola, si no llega el ACK se procede a hacer la retransmisión. El número máximo de retransmisiones para un paquete es de 30. En el caso de que el número de retransmisiones sea permanentemente grande, la cola se va a ir llenando. Cuando la cola tiene más de la mitad de paquetes en espera se dice que el nodo está congestionado. Para detectar paquetes duplicados el nodo evalúa el *SeqNo* y el origen del paquete y lo compara con un cache. Para detectar un loop el nodo evalúa el ETX de sus hijos, si el ETX es menor quiere decir que existe un loop dentro de la red. Ante un loop el procedimiento es activar un timer de BackOff y pedir una actualización de topología.

## 4.2. Creación de Clusters

Daca realiza la creación de clusters con el algoritmo Kmeans [10]. Kmeans es una técnica de cuatro pasos que se detalla en el algoritmo 1. Kmeans inicia al momento de posicionar  $k$  centroides uniformemente distribuidos en la red, posteriormente asigna los nodos a cada uno de los centroides de acuerdo a la mínima distancia entre el nodo y el centroide. La nueva posición de los centroides está determinada por el promedio en  $X$  y  $Y$  de los nodos pertenecientes al centroide. Si la posición del centroide cambia se ejecuta nuevamente el Kmeans, de lo contrario el algoritmo finaliza.

La figura 4.1 presenta el Kmeans de manera gráfica.

---

**Algorithm 1** Kmeans Algorithm

---

```
1:  $N$  is the number of nodes
2:  $K$  is the number of clusters
3:  $C_x = rand()$  and  $C_y = rand()$ 
4: for  $n < N$  do
5:    $ClosestDistance = \infty$ 
6:   for  $k < K$  do
7:     if  $Distance(C_{x,y}, Node) < ClosestDistance$  then
8:        $Cluster[n] = k$ 
9:        $ClosestDistance = Distance(C_{x,y}, Node)$ 
10:    end if
11:     $k = k + 1$ 
12:  end for
13:   $n = n + 1$ 
14: end for
15:  $C_{NewX} = Average(Nodes_x)$ 
16:  $C_{NewY} = Average(Nodes_y)$ 
17: if  $C_{NewX} \neq C_x \parallel C_{NewY} \neq C_y$  then
18:    $goto(2)$ 
19: else
20:    $End$ 
21: end if
```

---

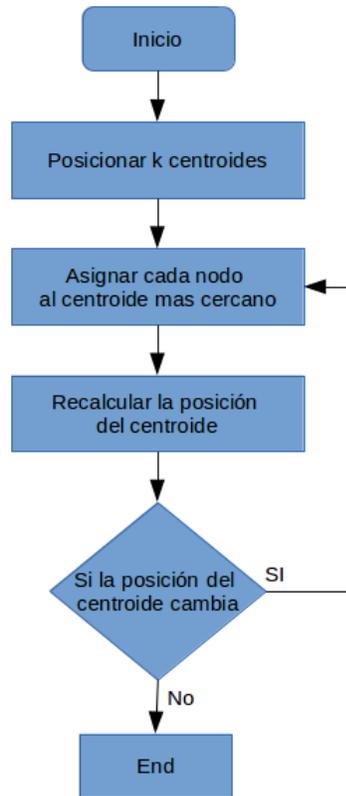


Figura 4.1: Algoritmo kmeans

### 4.3. Optimización Multi-Objetivo

La optimización multi-objetivo (MOO) es utilizada cuando se desea maximizar o minimizar más de una variable y adicionalmente estas variables son inversas, es decir, cuando una aumenta su valor la otra lo disminuye. Debido a que las WSN son restringidas en varios aspectos, es común que se presenten variables inversas, por lo tanto el uso de MOO es ampliamente difundido en esta área. El PSO y sus diferentes variantes han sido ampliamente usados en WSN [11] como técnicas MOO, la tabla 4.1 muestra algunas aplicaciones concretas.

El PSO es un método de optimización inspirado en la naturaleza que emula el comportamiento de parvada de aves o un cardumen de peces. El PSO fue planteado por primera vez por Kennedy y Eberhart [16] en 1995. El PSO inicia con un conjunto de soluciones arbitrario llamado la población inicial la cual se posiciona aleatoriamente en el hiperespacio del problema y cuyo movimiento está determinado por la ecuación de velocidad (4.1) y de posición (4.2). Una solución óptima al

Cuadro 4.1: Aplicaciones de PSO en WSN. Tabla tomada de [11].

Tarea del PSO	Criterio de Optimización	Algoritmo	Referencia
Posicionar Nodos Estacionarios	Max. Cobertura	PSO-Voronoi	[12]
Localizar Nodos	Min. Error de Localización	PSO-Loc	[13]
Seleccionar CH	Min. Distancia Intra-Cluster	PSO-Clustering	[14]
Encontrar Optima Potencia de TX	Min. Gasto de Energía y Error Prob.	PSO-Opt-Alloc	[15]

problema se busca actualizando la posición de cada una de las partículas.

Comparativamente el PSO tiene menos parámetros que los algoritmos GAs (Genetic Algorithm), no usa operadores evolutivos como mutación y crossover, y adicionalmente no requiere codificación y decodificación de números binario por lo cual resulta más fácil implementarlos en el computador [17].

EL PSO define los siguientes conceptos. Una partícula es un individuo de la manada que tiene una posición dentro del hiperespacio, cada una representa un punto de diseño que es una posible solución del problema. La posición de la partícula se da en coordenadas y su velocidad es la tasa a la cual se mueve. Y el *Swarm Leader* es la partícula con la mejor posición dentro de la manada. El movimiento de las partículas se define según las ecuaciones (4.1) y (4.2):

$$v_i(k+1) = \phi(k)v_i(k) + \alpha_1[\gamma_{1i}(p_i - x_i(k))] + \alpha_2[\gamma_{2i}(G - x_i(k))] \quad (4.1)$$

$$x_i(k+1) = x_i(k) + v_i(k+1) \quad (4.2)$$

Cuadro 4.2: Parámetros del Algoritmo PSO

Variable	Descripción
$i$	Índice de la partícula
$k$	Índice de tiempo discreto
$x$	Posición de la partícula $i$
$v$	Velocidad de la partícula $i$
$p$	Mejor posición de la partícula $i$
$G$	Mejor posición entre todas las partículas
$\gamma_{1,2i}$	Número aleatorio entre 0 y 1
$\alpha_{1,2}$	Constante de aceleración
$\phi$	Función de inercia

Los CH son seleccionados maximizando la función  $F$  a través de la técnica PSO. La función  $F$  se describe en la ecuación (4.3)

$$F = \max \left( \sum_{n=1}^K \frac{RE}{IE} + \frac{D}{MD} + C \right) \quad (4.3)$$

donde  $RE$  es la energía restante del nodo,  $IE$  es la energía inicial del nodo,  $D$  es la distancia del nodo al centroide,  $MD$  es la máxima distancia de un nodo al centroide dentro del cluster y  $C$  es el porcentaje de cobertura. Al maximizar la función  $F$  se eligen los CH con mayor energía, que estén más cerca al centroide y que garanticen la mayor cobertura. Cuando a un CH se le acaba la energía o falla, DACA vuelve a correr el PSO y selecciona otro CH que remplace al anterior.

#### 4.4. Creación de Caminos Disjuntos

Una vez se tengan los CH seleccionados a través del punto 4.3 se procede a realizar un nuevo árbol que solamente incluya los CH, esto con el fin de colocar los no-CH en modo inactivo para ahorrar energía y aumentar el tiempo de vida de la red. Posteriormente, se encuentran caminos disjuntos con la técnica N-to-1 [8].

La técnica *N-to-1 Multipath Routing* se compone de dos fases las cuales se explican a continuación:

1. *Branch Aware Flooding*: La forma general de un mensaje es  $mtype, mid, nid, bid, cst, path$ . La variable  $mtype$  identifica si el camino es primario(RPRI) o secundario, en esta fase del algoritmo los caminos siempre son primarios. La variable  $mid$  es el número de secuencia del mensaje. La variable  $nid$  es el identificador del nodo que envía el mensaje. La variable  $bid$  es el identificador de la rama, las ramas se identifican por el nodo que pertenece a esa rama que está más cerca del sumidero. La variable  $path$  contiene la secuencia de nodos por los cuales el mensaje a viajado. Y por último la variable  $cst$  es el costo del camino.

El sumidero inicia la actualización de las rutas enviando el siguiente mensaje de broadcast  $RPRI, mid, Sink, 0, 0, (Sink)$ , el cual significa que es una ruta primaria que conduce hacia el sumidero. Los nodos que escuchan este mensaje, por ejemplo  $z$ , marcan el sumidero como su padre y aprenden la ruta inversa  $path$ . Luego envían el siguiente mensaje  $RPRI, mid, z, z, cst + cst(z, parent(z)), path + z$  para seguir informando al resto de los nodos el camino correcto. Este proceso se repite.

Se tiene el concepto de nodos primos, los cuales son nodos vecinos en ramas distintas. Si llega un mensaje de actualización de un nodo primo que tiene un costo menor al camino primario, este camino se incluye dentro del conjunto de caminos disyuntos. De esta forma

se pueden tener caminos disyuntos como los que muestra la siguiente imagen. La imagen tiene un sumidero de color amarillo y tres ramas de color negro, azul y verde, las líneas rojas representan caminos disyuntos a través de nodos primos.

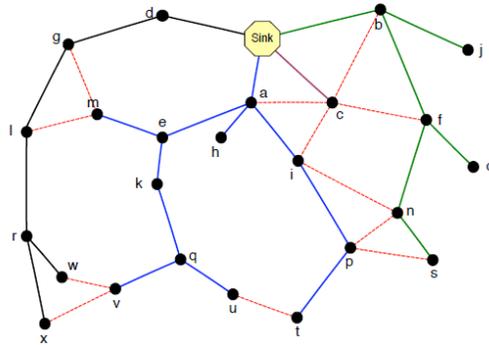


Figura 4.2: Árbol con Caminos Disyuntos. Tomado de [8].

2. *Multipath Extension Flooding*: La primera etapa del algoritmo encuentra caminos disyuntos únicamente a través de los nodos primos. Esta etapa encuentra más caminos disyuntos por medio de un segundo tipo de mensaje, RALT, que significa camino alterno. Cuando un nodo escucha un camino alterno verifica si éste proviene de su padre, si es así el mensaje se ignora. También se descarta el mensaje si el nodo mismo ya está incluido en el camino hacia el sumidero. Si las condiciones anteriores son falsas el nodo agrega la nueva ruta, el cuál es un camino disyunto que no es propio de un primo. Por ejemplo, en la figura anterior el nodo  $w$  publica un camino alterno  $w - v - q - k - e - a - sink$ , el cual es escuchado por  $r$  generándose el nuevo camino  $r - w - v - q - k - e - a - sink$ . Como se observa  $r$  y  $w$  no son primos, sin embargo, se pudo establecer un nuevo caminos disyunto. En esta segunda fase se encuentran nuevos caminos pero se agrega más tráfico de control a la red.

# Capítulo 5

## Análisis de Resultados

En este capítulo se presentan los parámetros de la simulación en la sección 5.1, las métricas usadas para evaluar DACA en la sección 5.2 y los resultados de la simulación en la sección 5.3.

### 5.1. Parámetros de la Simulación

En esta sección se muestran los parámetros escogidos para la simulación del algoritmo DACA. El simulador que se utilizó para el desarrollo de la tesis es Castalia una herramienta desarrollada por NICTA para evaluar protocolos de WSN que está basada en Omnet++. Se explican los parámetros de simulación según cada uno de los módulos de Castalia.

#### 5.1.1. Capa de Aplicación

La capa de aplicación en Castalia es el módulo de nombre *Application* en el cuál el desarrollador plantea su algoritmo a este nivel. En el caso del algoritmo DACA a través de esta capa el sink ejecuta los algoritmos Kmeans y PSO con el fin de seleccionar los CH de una manera centralizada. Los parámetros de cada uno de estos algoritmos se muestran en las tabla 5.1 y 5.2 respectivamente.

En el caso del kmeans se define el número de nodos activos  $k_{activos}$ , que es igual al número de clusters en la red. Este valor se define según la ecuación (5.1)

$$k_{activos} = \frac{AreaInteres}{\pi * R_{sensado}^2} \quad (5.1)$$

donde *AreaInteres* es el área total de la red y  $R_{sensado}$  es el radio de cobertura de cada nodo. El *AreaInteres* se eligió como 60x60 metros debido a que una longitud  $Field_{x,y}$  de 60 metros es una distancia conveniente de acuerdo a las potencias de transmisión que se están manejando, para mas detalle de este punto dirigirse a la capa de comunicación sección del radio 5.1.2. De otro lado

el radio de cobertura  $R_{sensado}$  se explica en la sección 5.1.3

Del mismo modo, para la función  $F$  se tiene que  $RE$  es el valor de la energía restante del nodo,  $IE$  es el valor de la energía inicial del nodo,  $D$  es la distancia del nodo al centroide,  $MD$  es la distancia máxima de un nodo al centroide dentro de un cluster específico y  $C$  es el porcentaje de cobertura de la red. Los parámetros  $RE$  y  $C$  de la función  $F$  son dinámicos, por lo tanto van cambiando en el tiempo. El número de nodos de la red se eligió en 100 para tener una densidad  $D$  de 1 nodo cada 36 metros cuadrados, ecuación (5.2).

$$D = \frac{\text{NumeroNodos}}{\text{AreaInteres}} \quad (5.2)$$

Cuadro 5.1: Parámetros Kmeans

Parámetro	Valor
AreaInteres	60x60 metros
Longitud en X y Y	$field_{x,y} = 60$ metros
$k_{activos}$	12
Función Multi-objetivo	$F = \max\left(\sum_{n=1}^K \frac{RE}{IE} + \frac{D}{MD} + C\right)$
$D$	$\frac{1}{36}$ Nodos por metro cuadrado

En el caso del PSO el parámetro  $n$  se refiere a la cantidad de partículas, este valor es igual a  $k$  debido a que dentro de cada cluster se posiciona una única partícula. Otro valor que se define es la función de inercia que por recomendación del paper [25] se define como una función lineal  $\phi$  que comienza con 0,9 y disminuye su valor hasta 0,4. El número de iteraciones  $nIteraciones$  se elige en 2000 para garantizar que el algoritmo PSO converja. De otro lado las constantes de aceleración  $\alpha_{1,2}$  se eligen en 2 según recomendación del paper [25].

Cuadro 5.2: Parámetros PSO

Parámetros	Descripción	Valor
$n$	Cantidad de Partículas	$k$
$\phi$	Función Inercia	$\phi = \frac{-0,5}{nIteraciones}x + 0,9$
$nIteraciones$	Número Iteraciones	2000
$\alpha_{1,2}$	Constante de Aceleración	2

### 5.1.2. Capa de Comunicación

La capa de comunicación en Castalia se divide en mac, radio y enrutamiento. El módulo de mac no se modificó debido a que no corresponde a los objetivos de esta tesis. Los parámetros del

radio y del enrutamiento se explican en la secciones que siguen.

### Radio

El radio que se utilizó en la tesis es el conocido transceptor CC2420. Este radio permite variar la potencia de transmisión y asignar una potencia de consumo correspondiente según la tabla 5.3. La potencia de recepción que viene por defecto es  $-95$  dBm con un consumo de 62 mW, sin embargo estos valores también son modificables.

Cuadro 5.3: Potencia de TX y consumo

Potencia de TX(dBm)	Consumo (mW)
0	57,42
-1	55,18
-3	50,69
-5	46,2
-7	42,24
-10	36,3
-15	32,67
-25	29,04

Como el algoritmo DACA permite activar o desactivar los nodos, es decir, ponerlos en modo sleep. Por lo tanto, se incluye en la tabla 5.4 el gasto de energía durante las transiciones entre RX, TX y sleep.

Cuadro 5.4: Matriz de Potencias de Transición(mW)

	RX	TX	SLEEP
RX	-	62	62
TX	62	-	62
SLEEP	1,4	1,4	-

Finalmente incluimos la gráfica 5.1 que muestra la distancia máxima de transmisión para cada una de las potencias de transmisión, el valor de la potencia de recepción se mantiene en  $-95$  dBm.

En la mayoría de simulaciones se usó una potencia de transmisión de  $-5$ dBm cuyo alcance es de 27 metros(ver gráfica 5.1), debido a esto el área de la red se eligió como 60x60 metros para obligar que entre los nodos y el sink existiera como máximo 3 saltos aproximadamente. Lo anterior

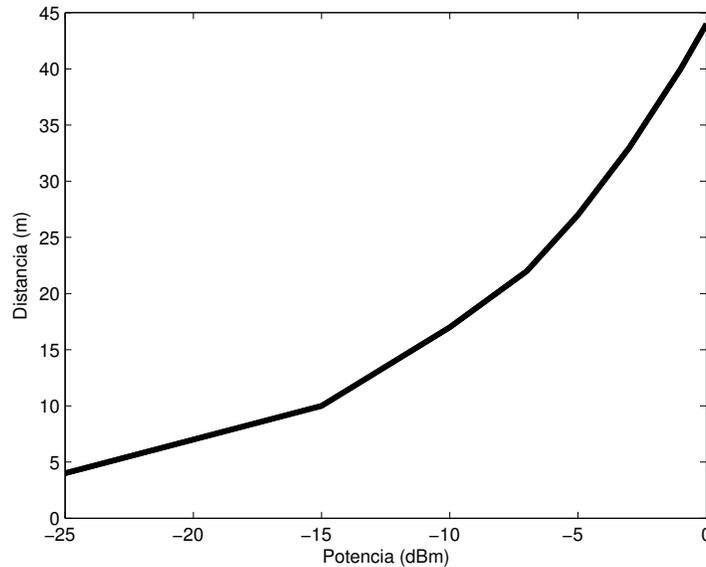


Figura 5.1: Distancia máxima de transmisión a diferentes potencias

con el fin de que cada rama no resultara con más de 3 nodos para poder evaluar más simplemente la generación de árboles con el CTP.

### Enrutamiento

El enrutamiento dentro de DACA se realiza con el algoritmo CTP del cuál resaltamos los parámetros más significativos para cada uno de sus módulos. En la tabla 5.5 se muestran los parámetros del RE. La variable  $I_b^{min}$  y  $I_b^{max}$  es útil para definir el tiempo mínimo y máximo que debe esperar un nodo para enviar un beacon. El tamaño de la tabla de enrutamiento es de 10 nodos vecinos, si se va a implementar una red con alta densidad de nodos se podría pensar en aumentar este valor. Cuando un nodo va a cambiar de padre debido a que otro padre mejor se presentó, entonces este nuevo padre debe demostrar que su valor ETX es menor que el ETX del padre anterior más el umbral de cambio de padre, este umbral es definido por CTP como 15. El procedimiento de selección de un padre se hace cada 8 segundos, esta variable se denomina tiempo de refrescamiento del padre. Y finalmente, el beacon tiene un tamaño de 5 bytes.

En la tabla 5.6 se muestran los parámetros del FE. El tamaño de la cola se establece como 12 paquetes. Los lazos de enrutamiento se detectan comparando el paquete que está a punto de transmitirse con el cache LRU que guarda la información de los últimos 4 paquetes transmitidos. La cantidad máxima de intentos de transmisión es de 30. También se tienen 4 relojes de backoff:  $TX - OK$  es el reloj que se inicia al momento de una transmisión exitosa para balancear el uso del

Cuadro 5.5: Parámetros CTP-RE

Parámetro	Descripción	Valor
$I_b^{min}$	Tiempo mínimo para enviar un mensaje de beacon	125 ms
$I_b^{max}$	Tiempo máximo para enviar un mensaje de beacon	500 s
$n$	Tamaño de la tabla de enrutamiento	10
$p$	Umbral de cambio de padre	15
$r$	Tiempo de refrescamiento del padre	8 s
$b$	Tamaño del mensaje beacon	5 bytes

canal por los diferentes nodos,  $TX - NOACK$  es el reloj que se inicia cuando no se recibe el ACK correspondiente,  $Congestion$  se inicia cuando el padre pone la bandera de congestion en 1 y  $Loop$  se inicia cuando un lazo de enrutamiento es detectado.

Cuadro 5.6: Parámetros CTP-FE

Parámetro	Descripción	Valor
$Q$	Tamaño de la cola	12 paquetes
$LRU$	Tamaño del cache LRU	4 paquetes
$MaxTX$	Cantidad máxima de intentos de TX	30
$TX - OK$	Backoff de transmisión correcta	15,6 – 30,3 ms
$TX - NOACK$	Backoff de transmisión sin ACK	15,6 – 30,3 ms
$Congestion$	Backoff de congestión del padre	15,6 – 30,3 ms
$Loop$	Backoff de lazo detectado	62,5 – 124 ms

En la tabla 5.7 se muestran los parámetros del LE. El valor de ETX es sometido a un filtro de suavizado, para lo cuál se utiliza la constante  $ALPHA$ . Cada  $w_b$  paquetes se calcula el valor de EXT, este valor es definido por CTP como 3 paquetes. Debido a que en la tabla de enrutamiento tenemos definidas 10 entradas, para cada una de esas entradas se necesita un valor de ETX, por lo tanto el valor de la tabla de ETX debe ser 10. Por último se definen el tamaño del encabezado, 2 bytes, y del *footer* < 45 bytes.

### 5.1.3. Manejo de Recursos

En el módulo de *ResourceManager* de Castalia tenemos definidas los parámetros de la tabla 5.8. Se define la energía inicial de cada nodo como 187,2 Joules, debido a que la energía inicial de unas pilas AA es de 18720 Joules, es decir, se elige 100 veces menos este valor debido a que se desea un tiempo de simulación corto que nos permita determinar el tiempo de vida de la red. Por el solo hecho del nodo estar vivo tiene un consumo base de 6 Joules, así el nodo no esté sensando, transmitiendo o recibiendo información. Este consumo base se calcula cada 1000 ms

Cuadro 5.7: Parámetros CTP-LE

Parámetro	Descripción	Valor
<i>ALPHA</i>	Constante de suavizado	0,9
<i>w<sub>b</sub></i>	Ventana para calculo de ETX	3 paquetes
<i>T</i>	Tamaño de la tabla ETX	10 paquetes
<i>Header</i>	Tamaño del encabezado	2 bytes
<i>Footer</i>	Tamaño del footer	< 45 bytes

según lo define el tiempo  $T$ . De otro lado, el radio de sensado del nodo  $R_{sensado}$  es de 10 metros por recomendación del paper [26]

Cuadro 5.8: Parámetros Manejo de Recursos

Parámetro	Descripción	Valor
<i>IE</i>	Energía Inicial	187,2 Joules
<i>Baseline</i>	Consumo base del nodo	6 Joules
<i>T</i>	Periodo de calculo de consumo	1000 ms
<i>R<sub>sensado</sub></i>	Radio de sensado del nodo	10 m

## 5.2. Métricas de la Simulación

Se definieron 3 métricas para medir los resultados de la simulación. Estas metricas son el *Riesgo*, la cantidad de nodos activos, la cantidad de nodos inactivos y el porcentaje de cobertura de la red.

### 5.2.1. Número de saltos hacia el sink

Para que un mensaje llegue al sink debe pasar por un número de nodos previamente. Este número de saltos depende de la potencia de transmisión de los nodos y del radio de sensado. Depende de la potencia de transmisión de los nodos debido a que una mayor potencia de transmisión permite alcanzar nodos más distantes, por lo tanto, reduciendo el número de saltos. Adicionalmente depende del radio de sensado debido a que con un menor radio de sensado se van a tener más nodos vecinos los cuales se van a convertir en padres, incrementando el número de saltos. Por lo anterior, durante la simulación se varia el radio de sensado y la potencia de transmisión de los nodos para encontrar una gráfica que nos muestre la relación entre estas variables.

### 5.2.2. Tiempo de Vida de la Red: Cantidad de Nodos Activos e Inactivos Vivos

Se considera que la red está con vida cuando existe al menos un CH que está enviando información al sink. Debido a esto necesario contar la cantidad de nodos activos e inactivos que aún tienen energía en la red. Por lo tanto, se define  $k_{inactivos}$  como la cantidad de nodos inactivos en la red que aún están con vida y  $k_{activos}$  se había definido previamente como la cantidad de nodos activos con vida de la red. Los nodos activos corresponden a los CH que activamente envían información al sink y los nodos inactivos son los no-CH que están en estado sleep.

### 5.2.3. Porcentaje de Cobertura

El porcentaje de cobertura de la red se mide de acuerdo a la ecuación (5.3), en la cuál si aumenta el número de nodos activos  $k_{activos}$  se va a tener una mejor cobertura. Sin embargo, el valor de  $k_{activos}$  depende del radio de sensado y el area de la red según la ecuación (5.1)

$$C = \frac{k_{activos} * \pi * R_{sensado}^2}{AreaInteres} \quad (5.3)$$

Adicionalmente, cabe resaltar que en el caso de que existan nodos los cuales sobrelapen su radio de sensado este valor se resta de la ecuación (5.3).

### 5.2.4. Mapa de Riesgo de Pérdida de Conectividad

El área de la red se divide en cuadrados más pequeños de 10x10 metros, dentro de estos cuadrados se suma la energía restante de los nodos activos y la energía restante de los nodos inactivos y al inverso de esta cantidad se le denomina *Riesgo*. La ecuación (5.4) muestra la métrica para el riesgo de pérdida de conectividad para un cuadrado  $i$ .

$$Riesgo_i = \frac{1}{RE_{activos} + RE_{inactivos}} \quad (5.4)$$

Si el *Riesgo* es un valor alto significa que la red está a punto de perder conectividad dentro de esa grilla, o si por el contrario el valor del *Riesgo* dentro de esa grilla es bajo entonces se dice que el riesgo de perder conectividad en esa zona es mínimo.

## 5.3. Resultados de la Simulación

En esta sección se evalúan los resultados de las simulaciones a través de las métricas explicadas en la sección 5.2. La sección 5.3.1 muestra cómo el algoritmo DACA genera clusters, árboles y caminos disjuntos cuando el radio de sensado y distribución inicial de los nodos varían. La sección 5.3.2 compara DACA y CTP a través del tiempo de vida y la cobertura que proporciona

cada uno. La sección 5.3.3 nos muestra el intercambio(*Tradeoff*) que existe entre la cobertura y el tiempo de vida de la red .La sección 5.3.4 muestra el mapa de riesgo de pérdida de conectividad generado en la red según la métrica definida en la ecuación(5.4).

### 5.3.1. Generación de Clusters, Árboles y Caminos Disjuntos

Se realiza la simulación para generar clusters, árboles y caminos disjuntos para tres casos diferentes variando la distribución inicial de los nodos en el área de interés y el radio de sentido, las tablas 5.9 5.10 5.11 muestra los parámetros para cada una de las simulaciones. La topología se genera por medio de una distribución uniforme con una semilla determinada, por lo tanto, en el parámetro topología de las tablas se encuentra la semilla que se utilizó para crear la topología. Adicionalmente, cabe resaltar que al modificar el radio de sentido  $R_{sensado}$  el parámetro  $k_{vivos}$  cambia según la ecuación (5.1), es decir, que al aumentar el  $R_{sensado}$ , el parámetro  $k_{vivos}$  disminuye.

Cuadro 5.9: **Parámetros de la Simulación Número 1**

Parámetro	Valor
<i>AreaInteres</i>	60x60 metros
$R_{sensado}$	5 metros
$k_{activos}$	46
Número de Nodos	100
Topología	<i>Seed</i> = 1
Potencia de TX	-5dBm
Potencia de RX	-95dBm
<i>IE</i>	187,2 Joules

Cuadro 5.10: **Parámetros de la Simulación Número 2**

Parámetro	Valor
<i>AreaInteres</i>	60x60 metros
$R_{sensado}$	10 metros
$k_{activos}$	12
Número de Nodos	100
Topología	<i>Seed</i> = 0
Potencia de TX	-5dBm
Potencia de RX	-95dBm
<i>IE</i>	187,2 Joules

Cuadro 5.11: **Parámetros de la Simulación Número 3**

Parámetro	Valor
<i>AreaInteres</i>	60x60 metros
<i>R<sub>sensado</sub></i>	15 metros
<i>k<sub>activos</sub></i>	5
Número de Nodos	100
Topología	<i>Seed</i> = 5
Potencia de TX	-5dBm
Potencia de RX	-95dBm
<i>IE</i>	187,2 Joules

DACA define los clusters haciendo uso del algoritmo Kmeans, construye los árboles por medio del algoritmo CTP y define los caminos disjuntos con la técnica N-to-1. En un primer momento se utilizan los **parámetros de la simulación 1** para generar los clusters, el resultado es la figura 5.2. Esta figura nos representa los nodos con el simbolo \*, los cuales están distribuidos de manera uniforme en el área de interés. La línea gruesa de color negro es la división de los clusters, en la figura 5.2 se pueden contar 46 clusters.

Kmeans nos garantiza que el promedio de la posición en  $X$  y el promedio de la posición en  $Y$  de cada uno de los centroides va a estar localizado aproximadamente en la mitad de la red (ecuación (5.5)), esto significa que los centroides van a estar uniformemente distribuidos en la red. Lo anterior significa que los radios de sensado de cada uno de los nodos van a estar uniformemente distribuidos, por lo tanto se va a minimizar el solapamiento de los mismos maximizando la cobertura.

$$average(C_{x,y}) \approx \frac{Field_{x,y}}{2} \quad (5.5)$$

Luego de que DACA separa los nodos en 46 clusters diferentes, se corre el algoritmo multi-objetivo PSO para escoger un CH dentro de cada cluster de acuerdo a la función de optimización (4.3). Los CH van a ser los únicos nodos activos en la red, debido a que los no-CH son puestos en modo inactivo. Con los CH seleccionados se crea un segundo árbol por medio del algoritmo CTP, recordemos que el primer árbol que DACA obtiene incluye todos los nodos de la red y es útil para volver el algoritmo centralizado enviando la información de la energía restante y posición de todos los nodos al sink.

El árbol construido únicamente con los CH se puede observar en rojo la figura 5.3. Cabe resaltar que este es el árbol generado durante la primera ronda de ejecución de DACA, es decir cuando aún no ha muerto ningún nodo. Sobre el árbol generado se corre el algoritmo N-to-1 para crear

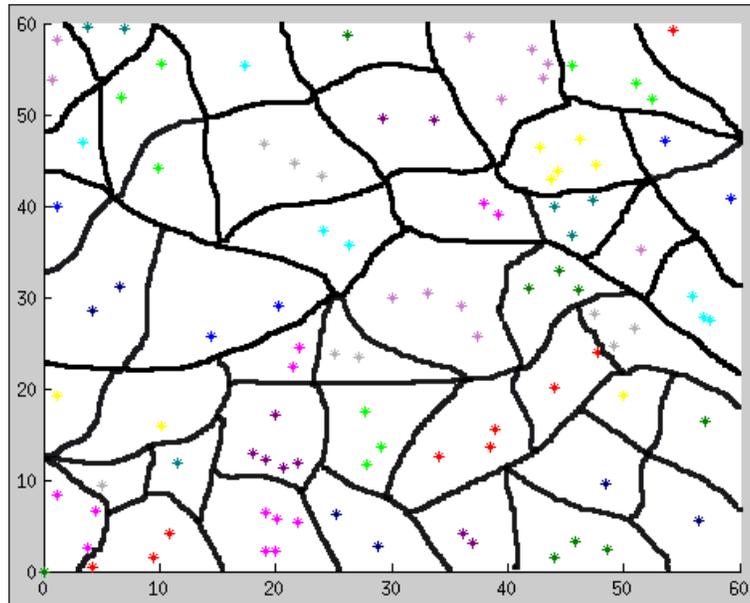


Figura 5.2: Kmeans N-Iteration para  $R_{sensado} = 5$  metros

los caminos disjuntos, los cuales se representan en la figura 5.3 con las flechas azules. Se puede observar una correspondencia entre las figuras 5.2 y 5.3 debido a que dentro de cada cluster de la figura 5.2 se elige un CH, el cuál es el único nodo de ese cluster que pertenece al árbol de la figura 5.3.

En la figura 5.3 podemos ver que el nodo 75 encuentra un camino disjunto bidireccional a través del nodo número 31. En caso de que el padre de 75 muera, este va poder seguir enviando información a través del nodo 31. De esta manera, la red se ha vuelto tolerante a fallos.

Se muestra ahora la generación de clusters, árboles y caminos disjuntos con los **parámetros de la simulación 2** donde se cambia la distribución de los nodos en el área de interés y el radio de sensado se hace mayor  $R_{sensado} = 10$ , lo cuál significa un menor número de clusters y un menor número de nodos activos  $k_{activos}$  según la ecuación (5.1). Nuevamente DACA genera los clusters con el algoritmo Kmeans dando como resultado la figura 5.4. Por la ecuación (5.1) vamos a tener un total de 12 clusters.

Posteriormente se corre PSO para seleccionar los CH dentro de cada cluster, maximizando la función (4.3). Con los CH seleccionados CTP crea un árbol rojo el cuál se muestra en la figura 5.5. Los caminos disjuntos son mostrados con las flechas azules. En la figura 5.5 podemos ver que el nodo 32 encuentra un camino disjunto bidireccional a través del nodo número 35. En caso de que el padre de 32 muera, este va poder seguir enviando información a través del nodo 35. De esta manera, la red se ha vuelto tolerante a fallos.

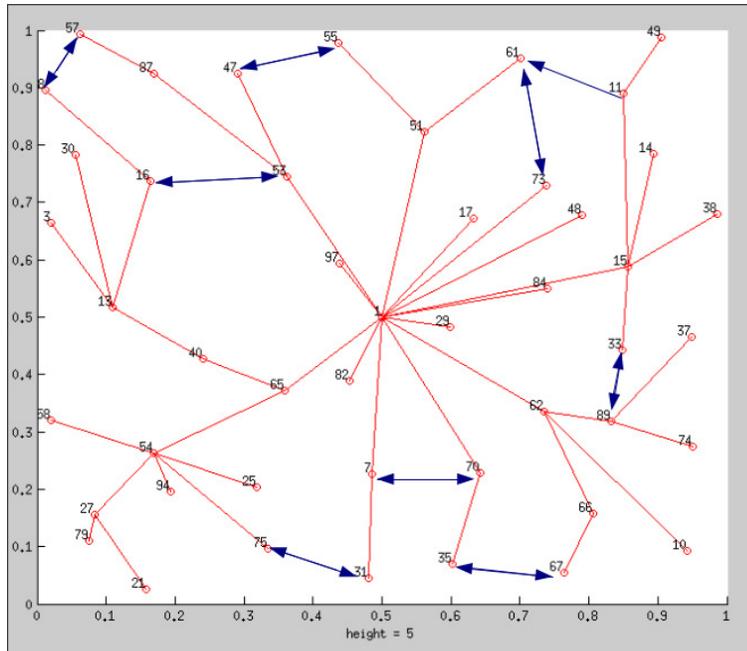


Figura 5.3: Árbol y caminos disjuntos para  $R_{sensado} = 5$  metros

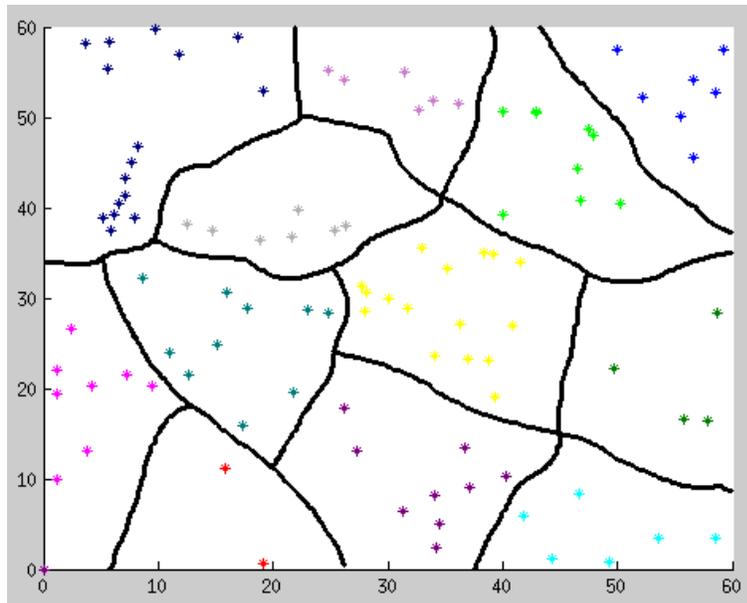


Figura 5.4: Kmeans N-Iteration para  $R_{sensado} = 10$  metros

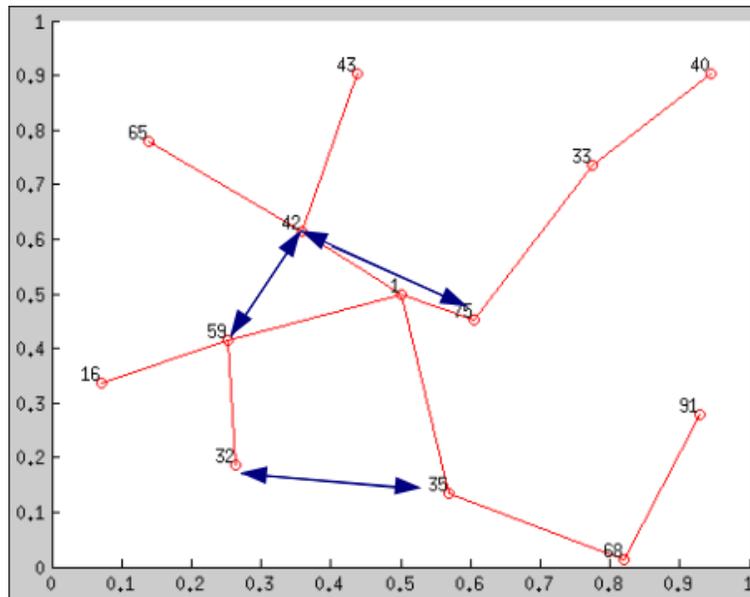


Figura 5.5: Árbol y Caminos Disjuntos para  $R_{sensado} = 10$  metros

Se muestra ahora la generación de clusters, árboles y caminos disjuntos con los **parámetros de la simulación 3** donde se cambia la distribución de los nodos en el área de interés y el radio de sensado se hace mayor  $R_{sensado} = 15$ , lo cuál significa un menor número de clusters y un menor número de nodos activos  $k_{activos}$  según la ecuación (5.1). Nuevamente DACA genera los clusters con el algoritmo Kmeans dando como resultado la figura 5.6. Por la ecuación (5.1) vamos a tener un total de 5 clusters.

Posteriormente se corre PSO para seleccionar los CH dentro de cada cluster, maximizando la función (4.3). Con los CH seleccionados CTP crea un árbol rojo el cuál se muestra en la figura 5.7. Los caminos disjuntos son mostrados con las flechas azules. En la figura 5.7 podemos ver que el nodo 82 encuentra un camino disjunto bidireccional a través del nodo número 95. En caso de que el link entre el sink(1) y 82 falle, el nodo 82 va poder seguir enviando información a través del nodo 95. De esta manera, la red se ha vuelto tolerante a fallos.

En esta sección evaluamos la generación de clusters, árboles y caminos disjuntos variando la distribución de los nodos en el área de interés y el radio de sensado. Vemos que el número de saltos hacia el sink depende del radio de sensado así como de la potencia de transmisión. Depende del radio de sensado ya que si éste es pequeño se va a contar con un gran número de nodos vecinos incrementando el número de saltos al sink. Adicionalmente, depende de la potencia ya que con

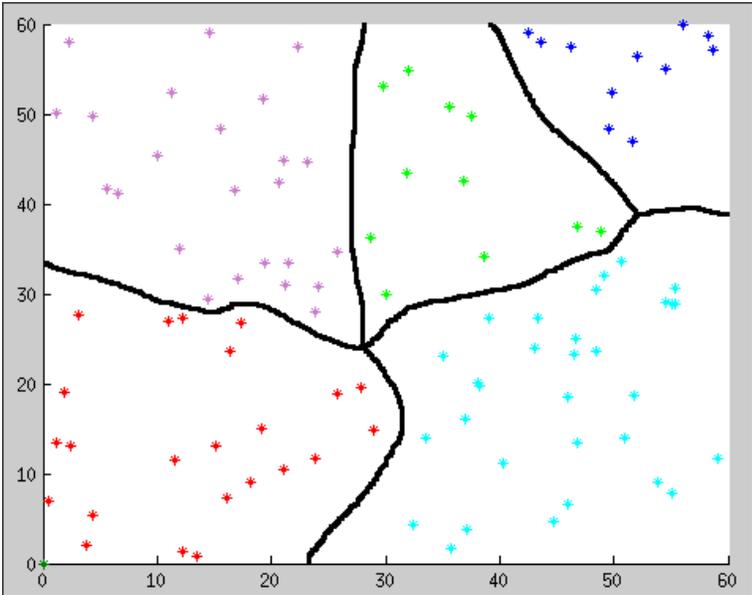


Figura 5.6: Kmeans N-Iteration para  $R_{sensado} = 15$  metros

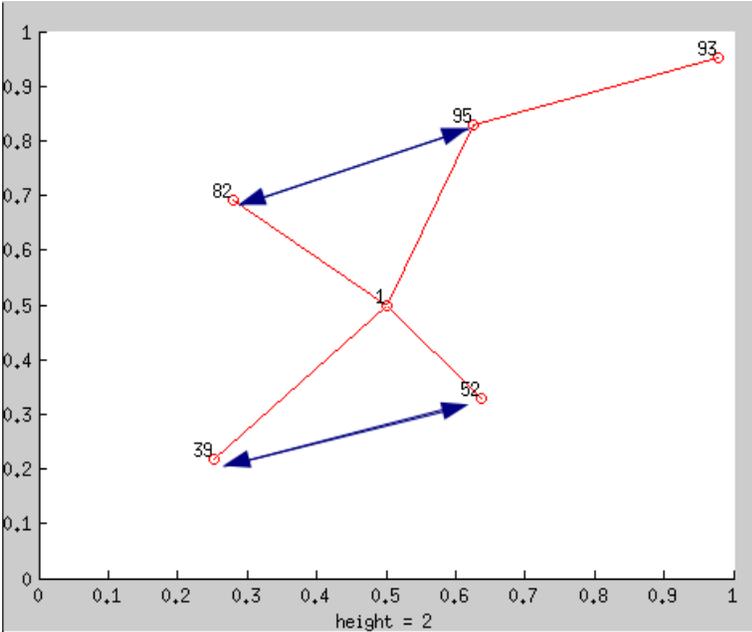


Figura 5.7: Árbol y Caminos Disjuntos para  $R_{sensado} = 15$  metros

una potencia de transmisión baja el mensaje debe saltar por varios CH antes de llegar al sink, y con una potencia de transmisión alta el mensaje puede llegar al sink hasta de un solo salto. La figura 5.8 muestra una gráfica con el radio de sensado en el eje  $x$  y el número de saltos en el eje  $y$ , la cuál se realiza para tres potencias de transmisión distintas.

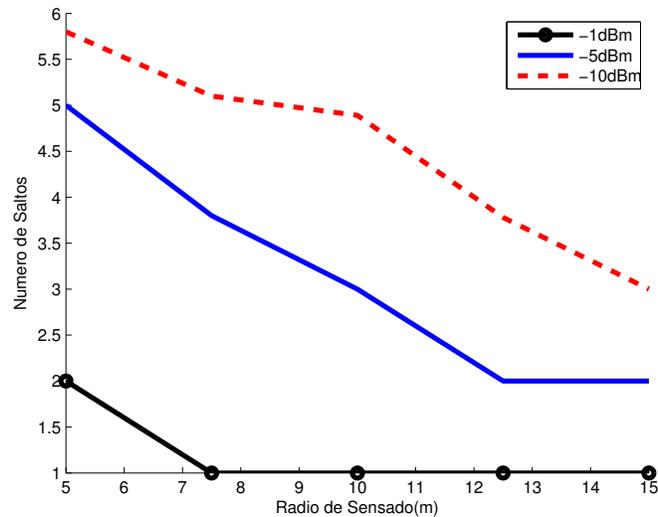


Figura 5.8: Número de saltos según la potencia y el radio de sensado

### 5.3.2. Comparación entre DACA y CTP en Cobertura y Tiempo de Vida

A nivel de cobertura y tiempo de vida de la red se compara DACA con el conocido algoritmo de control de topología CTP [7]. En DACA la totalidad de los nodos se puede dividir en activos e inactivos, debido a que esta técnica pone en estado inactivos algunos de ellos. De otro lado CTP no pone nodos inactivos, por lo tanto solamente van a existir nodos activos. Se realiza la comparación entre DACA y CTP utilizando los parámetros de la simulación de las tablas 5.9 5.10 5.11 descritas en la sección 5.3.1.

Con los **parámetros de la simulación 1** se genera la figura 5.9 que muestra la cantidad de nodos activos e inactivos en DACA y la cantidad de nodos activos en CTP en el tiempo. Se observa que a los 2752,5 segundos todos los nodos de CTP mueren debido a que se les termina la energía, mientras que en DACA los nodos agotan su energía a los 10478,18 segundos, esto significa que DACA aumenta el tiempo de vida de la red 3,8 veces. Este aumento en el tiempo de vida de la red se debe a que DACA inicia con 55 nodos en estado inactivo y sólo 45 en estado activo, lo cuál permite un ahorro de energía importante.

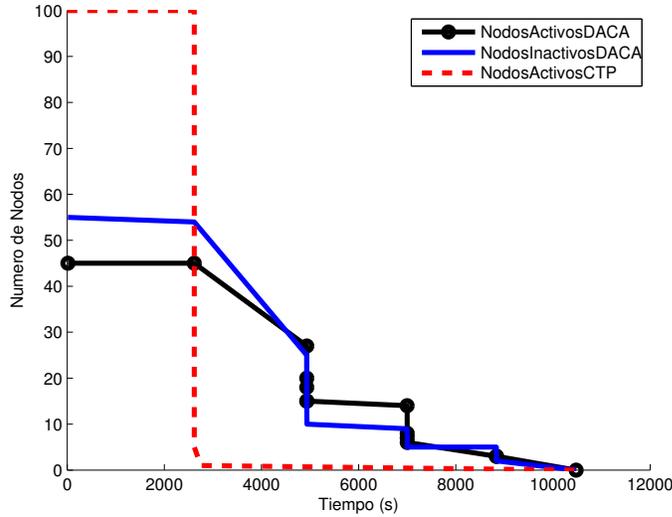


Figura 5.9: Comparación de nodos activos e inactivos entre DACA y CTP para  $R_{sensado} = 5$

De otro lado analizando la cobertura se genera la figura 5.10 que muestra el porcentaje de cobertura en el tiempo para DACA y CTP. Se observa que CTP mantiene una cobertura de 100 % por 2752,5 segundos y en este momento el sink deja de recibir información, es decir, la red se muere. De otro lado, la cobertura inicial en DACA es de 76,8 % y ésta va disminuyendo a medida que los nodos agotan su energía. Es decir, en DACA estamos aumentando el tiempo de vida de la red 3,8 veces, pero el costo de esto es que la cobertura no alcanza a ser 100 %, sino que inicia su valor máximo en 76,8 % y va disminuyendo su valor en el tiempo a medida que los nodos agotan su energía. El intercambio (*Tradeoff*) entre cobertura y tiempo de vida de la red se puede apreciar claramente en la sección 5.3.3.

Ahora bien, si modificamos el radio de sensado vamos a tener un cambio en el número de clusters y el número de nodos activos en la red (ecuación (5.1)), lo cuál afecta directamente el tiempo de vida de la red. Por esta razón se generan nuevamente las gráficas 5.9 5.10 pero con los parámetros de la simulación 2. La figura 5.11 muestra la cantidad de nodos activos e inactivos en DACA y la cantidad de nodos activos en CTP en el tiempo. Se observa que a los 2617,1 segundos todos los nodos de CTP mueren debido a que se les termina la energía, mientras que en DACA los nodos agotan su energía a los 20081,1 segundos, esto significa que DACA aumenta el tiempo de vida de la red 7,7 veces. Este aumento en el tiempo de vida de la red se debe a que DACA inicia con 88 nodos en estado inactivo y sólo 12 en estado activo, lo cuál permite un ahorro de energía importante.

Observamos que en la simulación con  $R_{sensado} = 5$  se aumentó el tiempo de vida de la red 3,8

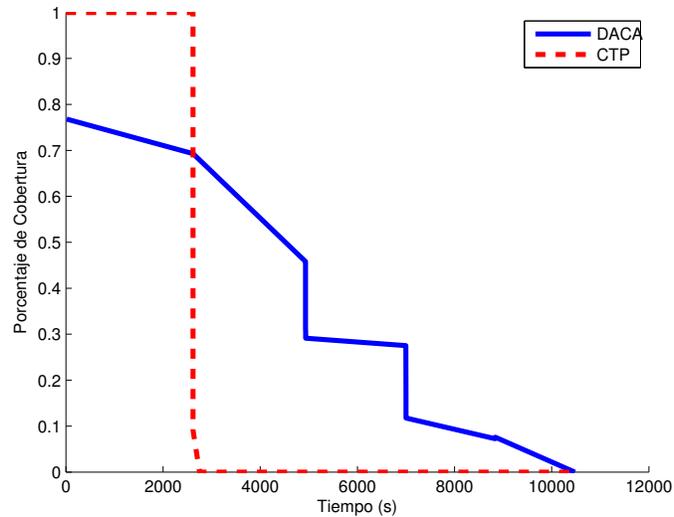


Figura 5.10: Comparación de cobertura entre DACA y CTP para  $R_{sensado} = 5$

veces, mientras que con  $R_{sensado} = 10$  se aumentó el tiempo de vida de la red en 7,7 veces, esto se debe a que entre mayor sea el radio de sensado, menor es el número de nodos activos que se necesitan para cubrir toda la red, lo cuál constituye un ahorro de energía de los nodos que permanecen en estado inactivo.

De otro lado analizando la cobertura se genera la figura 5.12 que muestra el porcentaje de cobertura en el tiempo para DACA y CTP. Se observa que CTP mantiene una cobertura de 100 % por 2617,1 segundos y en este momento el sink deja de recibir información, es decir, la red se muere. De otro lado, la cobertura inicial en DACA es de 78,6 % y ésta va disminuyendo a medida que los nodos agotan su energía. Es decir, en DACA estamos aumentando el tiempo de vida de la red 7,7 veces, pero el costo de esto es que la cobertura no alcanza a ser 100 %, sino que inicia su valor máximo en 78,6 % y va disminuyendo su valor en el tiempo a medida que los nodos agotan su energía. El intercambio (*Tradeoff*) entre cobertura y tiempo de vida de la red se puede apreciar claramente en la sección 5.3.3.

Por último generamos las gráficas 5.11 5.12 nuevamente pero esta vez con un radio de sensado de 15 para evaluar el comportamiento del tiempo de vida de la red y la cobertura. La figura 5.13 muestra la cantidad de nodos activos e inactivos en DACA y la cantidad de nodos activos en CTP en el tiempo. Se observa que a los 2617,4 segundos todos los nodos de CTP mueren debido a que se les termina la energía, mientras que en DACA los nodos agotan su energía a los 23140,4 segundos, esto significa que DACA aumenta el tiempo de vida de la red 8,8 veces. Este aumento

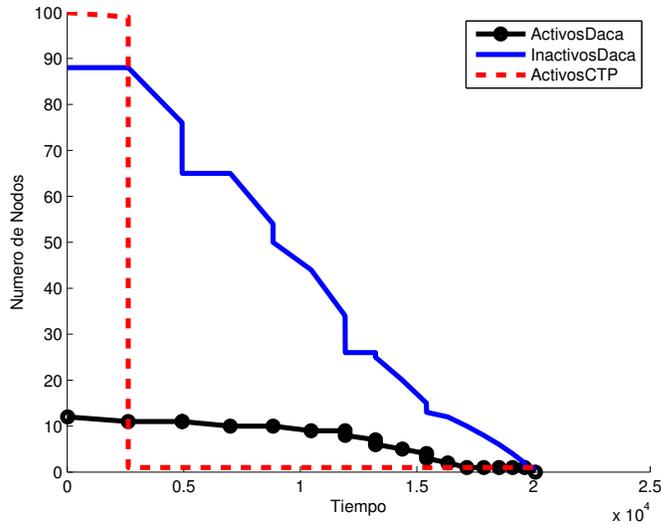


Figura 5.11: Comparación nodos activos e inactivos entre DACA y CTP en la red para  $R_{sensado} = 10$

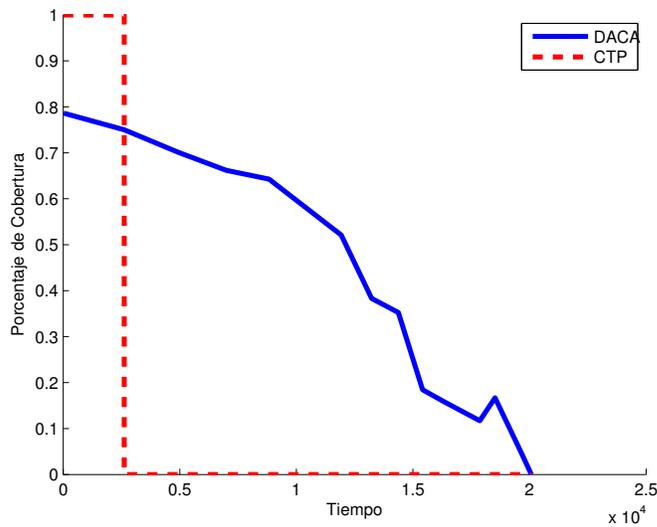


Figura 5.12: Comparación de cobertura entre DACA y CTP para  $R_{sensado} = 10$

en el tiempo de vida de la red se debe a que DACA inicia con 95 nodos en estado inactivo y sólo 5 en estado activo, lo cuál permite un ahorro de energía importante.

Observamos que en la simulación con  $R_{sensado} = 5$  se aumentó el tiempo de vida de la red 3,8

veces y con  $R_{sensado} = 10$  se aumentó 7,7, pero con  $R_{sensado} = 15$  el tiempo de vida de la red se incrementa drásticamente en 8,8 veces, esto se debe a que entre mayor sea el radio de sensado, menor es el número de nodos activos que se necesitan para cubrir toda la red y con  $R_{sensado} = 15$  la red únicamente necesita 5 nodos activos para cubrir toda el área de interés.

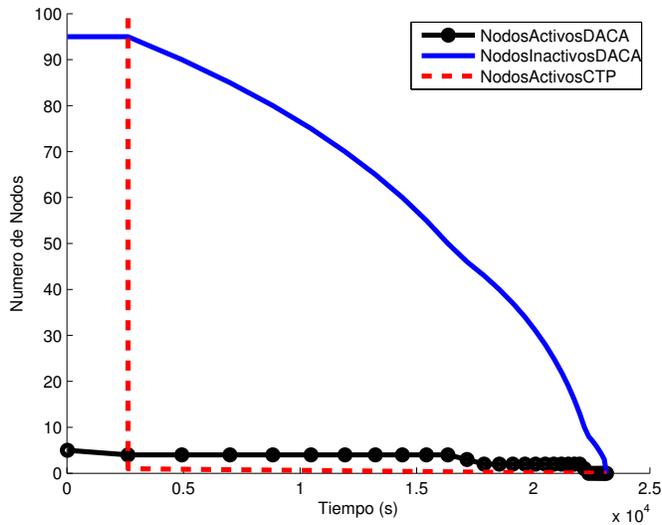


Figura 5.13: Comparación nodos activos e inactivos entre DACA y CTP en la red para  $R_{sensado} = 15$

El análisis de cobertura se realiza en la figura 5.14 que muestra el porcentaje de cobertura en el tiempo para DACA y CTP. Se observa que CTP mantiene una cobertura de 100 % por 2617,2 segundos y en este momento el sink deja de recibir información, es decir, la red se muere. De otro lado, la cobertura inicial en DACA es de 76,1 % y ésta va disminuyendo a medida que los nodos agotan su energía. Es decir, en DACA estamos aumentando el tiempo de vida de la red 8,8 veces, pero el costo de esto es que la cobertura no alcanza a ser 100 %, sino que inicia su valor máximo en 76,1 % y va disminuyendo su valor en el tiempo a medida que los nodos agotan su energía. El intercambio (*Tradeoff*) entre cobertura y tiempo de vida de la red se puede apreciar claramente en la sección 5.3.3.

### 5.3.3. Intercambio entre Cobertura y Tiempo de vida de la red

En la sección 5.3.2 vimos que el radio de sensado afecta considerablemente el tiempo de vida de la red ya que aumentándolo DACA puede incrementar hasta 8,8 veces el tiempo de vida de la red comparado con CTP. Sin embargo, al aumentar el radio de sensado estamos disminuyendo la granularidad de la medición de la cobertura. Por ejemplo, en la figura 5.15 se representan cuatro

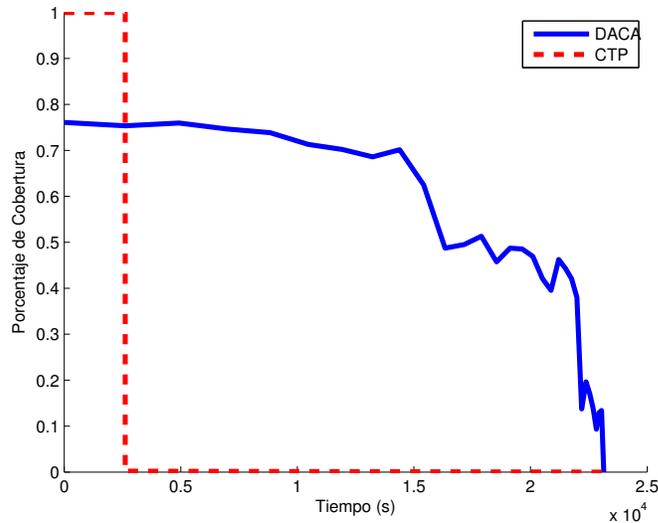


Figura 5.14: Comparación de cobertura entre DACA y CTP para  $R_{sensado} = 15$

nodos de color negro con un radio de sensado de 15 metros en un área de interés de color verde de  $60 \times 60$  metros, el área de interés que no está cubierta en la figura se muestran como los espacios de color verde. Si aumentamos la granularidad, disminuyendo el radio de sensado a 5 metros, nos encontramos con la figura 5.16. Claramente, la diferencia entra las figuras 5.15 5.16 es que con una mayor granularidad podemos cubrir mejor el área de interés.

Entonces, existe un intercambio(*Tradeoff*) entre el tiempo de vida de la red y la cobertura de la red, dado a que al aumentar el radio de sensado aumentamos el tiempo de vida de la red pero disminuimos la granularidad de la cobertura. La figura 5.17 nos muestra dos gráficas, donde la superior tiene en el eje  $x$  el radio de sensado y en el eje  $y$  tiene el porcentaje de cobertura, mientras que la inferior tiene en el eje  $x$  el radio de sensado y en el eje  $y$  el tiempo de vida de la red. Esta figura es útil ya que podemos observar claramente el intercambio entre porcentaje de cobertura y tiempo de vida de la red, dado que a medida que aumenta el radio de sensado el tiempo de vida de la red crece, pero el porcentaje de cobertura disminuye ya que la granularidad disminuye.

#### 5.3.4. Mapa Riesgo de Pérdida de Conectividad

El mapa de riesgo de pérdida de conectividad se realiza de acuerdo a la métrica definida en la ecuación (5.4), en la cuál se divide la red en grillas de  $10 \times 10$  metros y se suma la energía restante de los nodos activos y los nodos inactivos. Se calcula la métrica para DACA con los **parámetros de la simulación 2** y el resultado es la figura 5.18, en la cuál los puntos más altos representan un alto riesgo de pérdida de conectividad debido a que los nodos ya agotaron su energía en esa zona

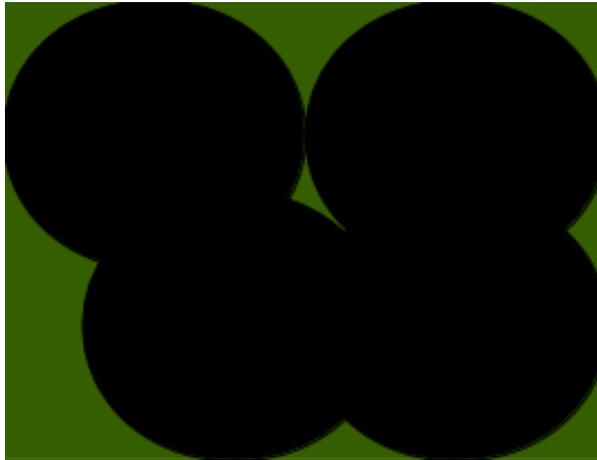


Figura 5.15: Granularidad con  $R_{sensado} \geq 15$

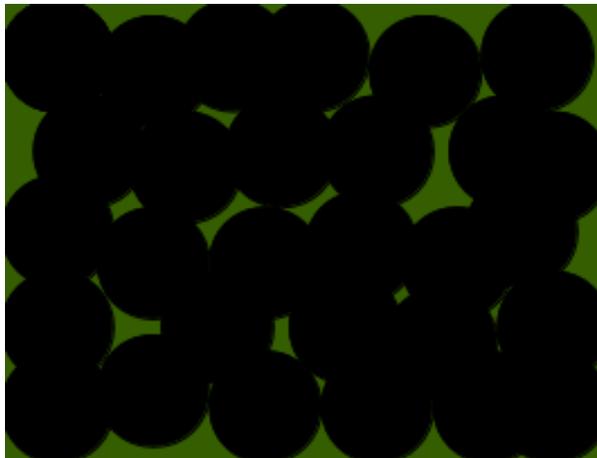


Figura 5.16: Granularidad con  $R_{sensado} \leq 5$

o debido a que no existen nodos en esa zona, mientras que los puntos bajos corresponden a zonas donde existen nodos que aún tienen energía restante.

De otro lado, se calcula el mapa de riesgo de pérdida de conectividad para CTP el cuál se muestra en la figura 5.19. La métrica  $Riesgo_i$  indica que DACA tiene 1,4 veces más energía disponible que CTP durante los primeros 400 segundos de simulación, por lo tanto, el riesgo de pérdida de conectividad para CTP es mayor, debido a que los nodos cuentan con menos energía.

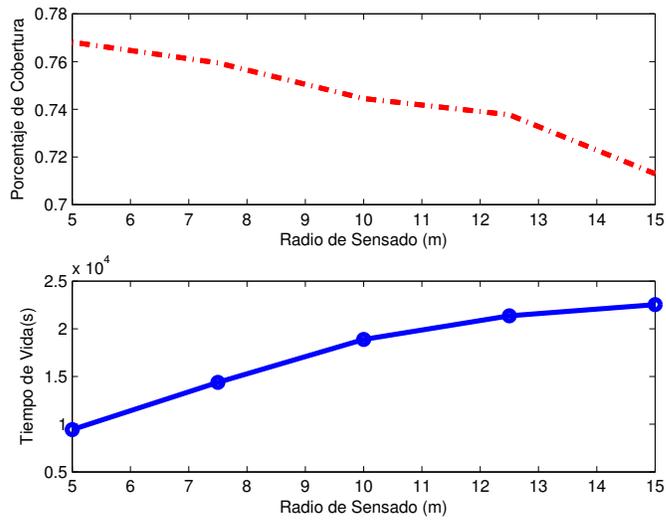


Figura 5.17: Intercambio entre el tiempo de vida de la red y el porcentaje de cobertura.

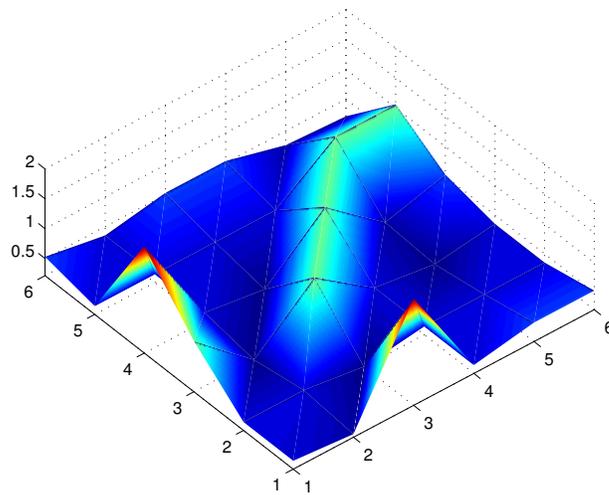


Figura 5.18: Mapa Riesgo de Pérdida de Conectividad para DACA

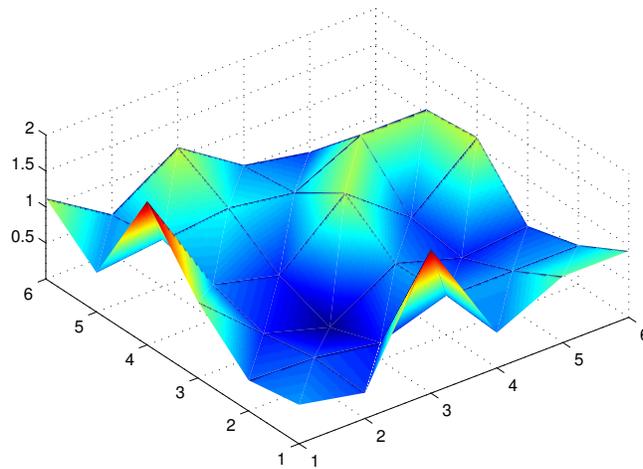


Figura 5.19: Mapa Riesgo de Pérdida de Conectividad para CTP

## Capítulo 6

# Conclusiones

En esta tesis se desarrolló la técnica denominada DACA (Disjoint path And Cluster Algorithm), el cuál es un algoritmo que incrementa el tiempo de vida de la red manteniendo la cobertura de la región de interés por medio de la auto-curación y la inactivación de nodos redundantes.

DACA se evaluó definiendo cuatro métricas diferentes. La primera de ellas es el **número de saltos hacia el sink**, en la cuál se analiza la relación que existe entre la potencia de transmisión y el radio de sensado al momento de determinar por cuántos saltos debe pasar un mensaje para llegar al sink. Los resultados muestran que entre menor sea el radio de sensado y mayor sea la potencia de transmisión seleccionada menor es el número de saltos hacia el sink. En WSN para ahorrar energía se desea que el número de saltos hacia el sink sea bajo pero igualmente se desea que la potencia de transmisión sea baja, por lo tanto, existe un intercambio (*tradeoff*) entre este par de variables. La evaluación de este intercambio corresponde a investigaciones que se realizarán posteriores a este trabajo de tesis.

La segunda métrica de evaluación de la técnica DACA es el **tiempo de vida de la red** que se establece como el tiempo durante el cuál al menos un CH envía información al sink. Los resultados muestran que DACA incrementa hasta 8,8 veces el tiempo de vida de la red respecto a CTP, debido a la técnica de auto-curación y a la utilización de CH que permite inactivar nodos dentro de cada cluster.

La tercera métrica de evaluación de DACA es el **porcentaje de cobertura**. Los resultados muestran que CTP mantiene una cobertura de 100 % de la red, sin embargo, el tiempo de vida de la red es muy corto como se mencionaba en la evaluación de la métrica anterior. De otro lado, DACA mantiene una cobertura de alrededor del 77 % durante casi 9 veces más tiempo que CTP. Es decir, DACA mantiene un buen nivel de cobertura durante mucho más tiempo que CTP.

Adicionalmente, los resultados de las simulaciones muestran que existe un **intercambio**(*tradeoff*) entre el tiempo de vida de la red y el porcentaje de cobertura. Cuando se incrementa el radio de sensado en las simulaciones, el tiempo de vida de la red crece debido a que existen más nodos inactivos en la red, pero el porcentaje de cobertura de la red disminuye debido a que la granularidad de la misma disminuye.

Por último, **el mapa de riesgo de pérdida de conectividad** define la métrica  $Riesgo_i$ , el cuál es útil para determinar en qué áreas de la red existe algún riesgo de pérdida de conectividad. Los resultados de la simulación muestran que DACA tiene menos riesgo de pérdida de conectividad que CTP debido a que pone los nodos en estado inactivo para ahorrar energía.

# Bibliografía

- [1] Y. Sankarasubramaniam F. Akyildiz, W. Su and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks, Volume 38, N. 4*, March 2002.
- [2] M. Francesco G. Anastasi, M. Conti and A. Passarella. Energy conservation in wireless sensor networks: a survey. *University of Pisa Italia*, 2007.
- [3] N. Peña D. Méndez and M. Guerrero. Flexible platform for wireless sensor networks. *Circuit Cellar: The Magazine for Computer Applications. Ed 211*, February 2008.
- [4] N. Peña D. Méndez and M. Guerrero. Wireless sensor networks: Structure and application. *Science and Innovation Magazine. Colombian Society for the Advance of Science. Vol XIV. Special Edition N° 3*, 2007.
- [5] G. Bacco F Coumo and T. Melodia. Shaper: A self-healing algorithm producing multihop bluetooth scatternets. *IEEE Globecom*, 2003.
- [6] C. Petrioli and S. Basagni. Degree-constrained multihop scatternet formation for bluetooth networks. *NorthEastern University IEEE*, 2002.
- [7] U. Colesanti and S. Santini. The collection tree protocol for the castalia wireless sensor networks simulator. *Technical Report Nr. 729, Department of Computer Science ETH Zurich*, June 2011.
- [8] W. Lou. An efficient n-to-1 multipath routing protocol in wireless sensor networks. *IEEE, Worcester Polytechnic Institute*, 2005.
- [9] K. Jamieson S. Kim P. Levis R. Fonseca, O. Gnawali and A. Woo. The collection tree protocol. *Network Working Group TinyOS Community*, February 2007.
- [10] P. Sasikumar and S. Khara. K-means clustering in wireless sensor networks. *IEEE, Fourth International Conference on Computational Intelligence and Communications Networks*, 2012.
- [11] R. Kulkarni and G. Kumar. Particule swarm optimization in wireless sensor networks: A brief survey. *IEEE Transactions on System, MAN and Cybernetics - Part C: Applications and Reviews*, March 2011.

- [12] A.W. Mohemmed N. A. B. A. Aziz and B. S. D. Sagar. Particle swarm optimization and voronoi diagram for wireless sensor networks coverage optimization. *Proc. Int. Conf. Intell. Adv. Syst.*, 2007.
- [13] A. Gopakumar and L. Jacob. Localization in wireless sensor networks using particle swarm optimization. *Proc. IET Int. Conf. Wireless, Mobile Multimedia Netw.*, 2008.
- [14] S. Halgamuge S. Guru and S. Fernando. Particle swarm optimisers for cluster formation in wireless sensor networks. *Proc. Int. Conf. Intell. Sens., Sens. Netw. Inf. Process.*, S. K. Halgamuge, 2005.
- [15] T. Wimalajeewa and S. K. Jayaweera. Optimal power scheduling for correlated data fusion in wireless sensor networks via constrained pso. *IEEE Trans. Wireless Commun.*, Sep. 2008.
- [16] J. Kennedy and R. Eberhart. Particule swarm optimization. purdue school of engineering and technology. *IEEE, Indianapolis*, 1995.
- [17] J. Arora. *Introduction to Optimum Design*. Elsevier, Third Edition, 2002.
- [18] S. Wang X. Wang and D. Bi. Dynamic sensor node selection strategy for wireless sensor networks. *Internacional Symposium on Communications and Information Tecnologies*.
- [19] C. Chiang K. Shih, Y. Chen and B. Liu. A distributed active sensor selection scheme for wireless sensor networks. *IProceedings of the IEEE Symposium on Computers and Communications*, June 2006.
- [20] P. Wightman and M. Labrador. A3: A topology construction algorithm for wireless sensor networks. *Department of computer Science and Engineering, University of South Florida. IEEE*, 2008.
- [21] N. Latiff. Energy-aware clustering for wireless sensor networks using particle swarm optimization. *The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2007.
- [22] M. Maimour. Maximally radio-disjoint multipath routing for wireless multimedia sensor networks. *Nancy University, France. ACM*, October 2008.
- [23] S. Shenker D. Ganesan, R. Govindan and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor network. *University of California, LA. ACM*, 2001.
- [24] D. Tipper X. Hou and J. Kabara. Label-based multipath routing (lmr) in wireless sensor routing. *In Proceedings of the 6th International Symposium on Advanced Radio Technologies (ISART 04), Boulder, CO*, 2004.
- [25] B. Birge. Psot - a particule swarm optimization toolbox for use with matlab. *NCSU, MAE Dept, IEEE*, 2003.

- [26] Y. Tao and L. Yuan. Optimized coverage algorithm in wsn based on energy balance. *IEEE, Chongqing University of Posts and Telecommunications*, 2008.