



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

Universidad Nacional Autónoma de México

Posgrado en Ciencia e Ingeniería de la Computación

**”Métodos eficientes para la recolección de datos en Redes
Inalámbricas de Sensores”**

Que para optar por el grado de:
DOCTORA EN INGENIERÍA (COMPUTACIÓN)

PRESENTA:

Martha Ma. Montes de Oca Cáliz

TUTOR:

Dr. Javier Gómez Castellanos, Facultad de Ingeniería

México, D.F., Marzo 2014



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

A Papá

Porque también somos lo que hemos perdido

A Mamá

Por ser como una lámpara de inagotable aceite
y mi guerrera favorita

A mis hermanas, hermanos y sobrinos

Porque si estamos juntos nada nos detendrá

”Y cuando tomamos conciencia de que
el imposible es eso: un imposible,
ya es tarde para refugiarnos en
la sensatez.” M.B.

Agradecimientos

De repente uno se pone a pensar quiénes estuvieron involucrados en la culminación de este trabajo y resulta que la lista es larguísima. Les agradezco a TODOS y ofrezco una disculpa a aquellos en los que pensé pero no pude escribir debido a la naturaleza breve de este espacio. Agradezco infinitamente:

Al Dr. Javier Gómez Castellanos por su paciencia y enseñanzas y por enderezar lo que estaba torcido desde el inicio.

A Tor porque está mirando el mar y el mar lo mira a él. Su esencia es tan sólida que abruma.

Al Dr. Imanol Ordorika Sacristán por tolerar mis ausencias sabiendo que con trabajo construía una presencia, y por supuesto a la DGEI por el constante apoyo.

Al Dr. Miguel López Guerrero por brindarme su tiempo y compartir conmigo la precisión que lo caracteriza.

Al Dr. Michael porque encontró lo que otros buscaron y no vieron.

A los Dres. Francisco García Ugalde y Víctor Rangel Licea por robustecer este trabajo y en especial al Dr. José Antonio García Macías por aquellas aportaciones que vencieron las distancias.

Al Dr. Andrew T. Campbell por mostrarme luz donde había oscuridad.

A Mamá Lupita por ser el origen. A Migue y Lichito por su magia. A Raúl, Blanquis, Sandy (y anexas) y Mariela porque llevamos agua al desierto. A Naye y Adri por sus sonrisas en los días difíciles.

Finalmente quiero dar las gracias a la UNAM por enaltecer los espíritus, al Posgrado en Ciencia e Ingeniería de la Computación y al CONACYT porque sus depósitos en tiempo siempre me alimentaron.

Resumen

Las redes inalámbricas de sensores son una tecnología integrada por un conjunto de nodos sensores que tienen la capacidad de auto organizarse, monitorear algún fenómeno de interés y transmitir los datos adquiridos a través de una red inalámbrica hacia un nodo central o hacia sus vecinos. Una de las tareas más importantes en este tipo de redes es la recolección de datos para obtener información completa de lo que sucede con determinado fenómeno. Debido a que el mayor gasto de recursos ocurre durante la transmisión y recepción de la información, es necesario el desarrollo de métodos eficientes que permitan conocer toda la información de la red y lograr un ahorro de recursos. En este trabajo se presentan dos métodos para la recolección de datos en las redes WSN, el primero es *Disagree*, el cual es un método que explota la correlación espacial de los datos al nivel del sensor gracias a que la solicitud de información se ajusta a los valores de cada vecindario (*cluster*). Más que solicitar a todos los nodos el envío de información para obtener una vista completa de la red, *Disagree* solicita información a aquellos sensores cuyas lecturas estén fuera de un intervalo centrado en la lectura reportada por el nodo árbol (*cluster head*). Las redes WSN integradas por sensores estáticos no son fácilmente escalables, por ello para el monitoreo de áreas muy extensas se recomienda el uso de sensores móviles. El segundo método es *Urbihoc*, el cual permite la recolección de datos en áreas grandes, no utiliza infraestructura costosa ya que cada nodo móvil adquiere información global del comportamiento del fenómeno a partir de intercambios locales con otros nodos móviles. En *Urbihoc*, un nodo móvil puede ser un vehículo particular, una unidad de transporte público o un teléfono celular. Los datos intercambiados pueden estar en modo resumido o en modo detallado, el modo resumido permite además un ahorro de energía pues se disminuye la cantidad de información a transmitir.

Lista de Acrónimos

- APTEEN** Adaptive Periodic Threshold-sensitive Energy Efficient
- CDS** Connected Dominating Set
- CAG** Clustered AGregation
- CODA** Congestion Detection and Avoidance
- CSMA/CA** Carrier Sense Multiple Access with Collision Avoidance
- DCS** Data-Centric Storage
- DDCHS** Density and Distance based Cluster Head Selection
- EEDCP-TB** Energy Efficient Data Collection Protocol - Tree Based
- ES** External Storage
- GPS** Global Positioning System
- HEED** Hybrid Energy Efficient Distributed
- H-HEED** Heterogeneous-Hybrid Energy Efficient Distributed
- IEEE** Institute of Electrical and Electronics Engineers
- ITS** Intelligent Transportation Systems
- LEACH** Low-Energy Adaptive Clustering Hierarchy
- LS** Local Storage
- MRCQ** Multi Resolution Compression and Query framework
- MST** Minimum Spanning Tree
- SHRP** Simple Hierarchical Routing Protocol
- SMAC** Sensor Medium Access Control Protocol
- SMRDD** Spatial-based Multi-Resolution Data Dissemination

TDMA Time Division Multiple Access

TEEN Threshold sensitive Energy Efficient

TCP Transport Control Protocol

WSN Wireless Sensor Network

Índice general

1	Introducción	1
1.1	Antecedentes	1
1.2	Planteamiento del problema	13
1.3	Objetivos	15
1.4	Contribución y relevancia de la tesis	15
1.5	Estructura de la tesis	16
2	Conceptos fundamentales y estado del arte	18
2.1	Conceptos fundamentales	18
2.2	Características de las WSN	19
2.2.1	Implementación e infraestructura	21
2.2.2	Topología	22
2.2.3	Protocolos de comunicación	23
2.2.4	Tipos de cobertura	26
2.2.5	Clasificación de las aplicaciones	26
2.3	Recolección de datos y estado del arte	28
3	Método <i>Disagree</i>	34
3.1	Introducción	34
3.2	Trabajo relacionado	38
3.3	Descripción de <i>Disagree</i>	43
3.3.1	Estructura	44
3.3.2	Expansión	44
3.3.3	<i>Disagree</i>	45
3.3.4	Respuesta	47

3.3.5	Rotación	49
3.4	El valor de Δ	51
3.5	Explotando la correlación espacial	52
3.6	Implementación, simulaciones, experimentos y resultados	56
3.6.1	Simulaciones	58
3.6.2	Experimentos	69
3.7	Conclusiones del capítulo	71
4	Método <i>Urbihoc</i>	72
4.1	Introducción	72
4.2	Formas de recolección de datos	74
4.3	Retos	75
4.4	Trabajo relacionado	77
4.5	Urbihoc	80
4.5.1	Clasificación	82
4.5.2	Intercambio del conocimiento resumido (S)	83
4.5.3	Intercambio de solicitudes y conocimiento detallado	84
4.5.4	Cantidad de información almacenada	85
4.5.5	Estimación de datos	86
4.6	Evaluación del desempeño	86
4.6.1	Simulaciones	87
4.6.2	Implementación	93
4.7	Conclusiones del capítulo	96
5	Conclusiones y trabajo futuro	98
A	Lista de publicaciones	101

Índice de figuras

1.1	(a)Arquitectura móvil donde los sensores utilizan la infraestructura de la telefonía celular y puntos de acceso WiFi para enviar sus datos. (b)Arquitectura móvil donde los sensores utilizan puntos de acceso WiFi para enviar sus datos. (c)Arquitectura móvil donde los sensores no utilizan ningún tipo de infraestructura y se organizan en modo ad hoc para enviar sus datos.	4
1.2	<i>Hexagonal grid</i>	5
1.3	Escalabilidad	6
1.4	Imagen que representa el fenómeno a monitorear.	7
1.5	(a)(b) y (c) Escenarios monitoreados por 4, 9 y 12 sensores estáticos. (d)(e) y (f) Resultados de solicitar una vista completa de los escenarios (a) (b) y (c) respectivamente.	8
1.6	(a)Los sensores se mueven siguiendo rutas predeterminadas. (b)Los sensores se mueven con base en rutas no predeterminadas.	8
1.7	(a)Sensores estáticos con información actualizada para cada tiempo t_i . (b)Sensores móviles que sólo tienen información actualizada de determinadas zonas en el tiempo t_i	10
1.8	(a)Arquitectura móvil que utiliza infraestructura para enviar datos hacia el nodo <i>sink</i> . (b)Arquitectura móvil que utiliza infraestructura para enviar y consultar información de una base de datos centralizada. (c)-(d)Después de los movimientos, un nodo queda fuera de la cobertura de la telefonía celular y los puntos de acceso WiFi.	11
1.9	(a)El nodo A envía datos al nodo <i>sink</i> a través de una ruta multisalto. (b)Después de reorganizarse, el nodo A nuevamente envía datos al nodo <i>sink</i> a través de una nueva ruta. (c)Los nodos se mueven y se reorganizan nuevamente pero dos nodos quedan desconectados del resto.	13

1.10	(a)Ejemplo de red donde los nodos sólo comparten información con los vecinos que se encuentran a un salto. (b)Después de que se mueven a otro lugar, los nodos pueden seguir compartiendo información con sus vecinos actuales.	13
2.1	Tipos de dispositivos con sensores	18
2.2	Componentes clave de los sensores	19
2.3	(a) El sensor A envía sus lecturas obtenidas al nodo <i>sink</i> (transmisión multisalto). (b) El sensor A comparte sus lecturas obtenidas con los nodos cercanos (transmisión de un salto).	20
2.4	Protocolos de comunicación utilizados por las redes WSN.	23
2.5	(a) Cobertura tipo manta. (b) Cobertura de objetivo. (c) Cobertura de barrera.	27
3.1	Soluciones para la adquisición de datos en las redes inalámbricas de sensores	37
3.2	Etapa de estructura	45
3.3	Etapa de expansión	46
3.4	Etapa <i>Disagree</i>	47
3.5	Etapa de Respuesta	48
3.6	Reconstruyendo una vista completa del comportamiento del fenómeno de interés	49
3.7	Rotación del árbol. (a) Un árbol en la época uno y (b) El árbol resultante de aplicar el algoritmo de rotación.	51
3.8	(Superior)Una red de sensores sobre un campo monitoreado. La superficie representa los valores del fenómeno en la red, las dos curvas de nivel están indicadas con $f(x, y) = Z_{TN} + \Delta$ y $f(x, y) = Z_{TN} - \Delta$. (Inferior) Nos enfocamos en TN_3 y hacemos un acercamiento alrededor de éste. Se puede observar cómo las dos curvas de nivel establecen áreas blancas y grises sobre la cobertura de TN_3	54
3.9	Áreas de desacuerdo (gris) para (a) $\Delta = 1$, (b) $\Delta = 5$, y (c) $\Delta = 10$. . .	55
3.10	Porcentaje de nodos que transmitieron y los que no transmitieron	56
3.11	Escala de lectura de datos. Escenarios (a) Mosaic, (b) Gaussian y (c) Random	57

3.12 Paquetes transmitidos: Escenario <i>Gaussian</i> con intervalos de confianza del 95 %.	60
3.13 Paquetes transmitidos: Escenario <i>Mosaic</i> con intervalos de confianza del 95 %.	60
3.14 Paquetes transmitidos: Escenario <i>Random</i> con intervalos de confianza del 95 %.	61
3.15 Porcentaje de nodos que no transmitieron: Escenario <i>Gaussian</i> con intervalos de confianza del 95 %.	62
3.16 Porcentaje de nodos que no transmitieron: Escenario <i>Mosaic</i> con intervalos de confianza del 95 %.	63
3.17 Porcentaje de nodos que no transmitieron: Escenario <i>Random</i> con intervalos de confianza del 95 %.	63
3.18 Error de estimación: Escenario <i>Gaussian</i> con intervalos de confianza del 95 %.	64
3.19 Error de estimación: Escenario <i>Mosaic</i> con intervalos de confianza del 95 %.	65
3.20 Error de estimación: Escenario <i>Random</i> con intervalos de confianza del 95 %.	65
3.21 Número de <i>TN</i> (nodos árboles) y <i>SN</i> (nodos sensores)	66
3.22 Comparación de escenarios	67
3.23 Rotación del rol del nodo árbol	67
3.24 Número de nodos vivos	69
3.25 Experimentos	70
4.1 (a) Imagen que representa el fenómeno a monitorear. (b) Datos adquiridos por un nodo móvil durante una jornada. (c) Datos adquiridos por el mismo nodo móvil gracias a los intercambios realizados con otros nodos móviles utilizando WiFi en modo ad hoc.	73
4.2 Frecuencia de monitoreo del fenómeno de interés y el tiempo de información	77
4.3 Trayectoria de dos nodos móviles	81
4.4 Cada nodo difunde un paquete p_a que contiene su ID y su conocimiento resumido S .	83
4.5 (a) Conocimiento adquirido por el nodo identificado con el número 50. (b) Conocimiento detallado ($b = 1$) adquirido por el nodo 50. (c) Conocimiento interpolado del nodo 50 a partir del conocimiento detallado.	87

4.6	(a) Imagen utilizada para representar el fenómeno, sobre ésta se colocaron aleatoriamente 100 nodos móviles. (b) Información adquirida por el nodo identificado con el número 50. (c) Conocimiento resumido (S) adquirido por el nodo 50 utilizando $b = 5$. (d) Interpolación del conocimiento resumido. (e) Conocimiento detallado (Z) ($b = 1$) adquirido por el nodo 50. (f) Interpolación del conocimiento detallado.	89
4.7	Porcentaje de similitud entre la información original y la información obtenida.	90
4.8	Número de bytes transmitidos, recibidos y total de conocimiento resumido (S) ($b = 5$) por el nodo 50.	90
4.9	Número de bytes transmitidos, recibidos y total de conocimiento detallado (Z) por el nodo 50.	91
4.10	(a) Conocimiento resumido con $b = 5$ y su interpolación (b). (c) Conocimiento resumido con $b = 10$ y su interpolación (d). (e) Conocimiento resumido con $b = 15$ y su interpolación (f).	92
4.11	Trayectoria del usuario con el nodo móvil identificado con el número 1.	93
4.12	Conocimiento adquirido individualmente por el nodo 1 durante toda su trayectoria.	94
4.13	Parte de las trayectorias de los usuarios 1 y 2.	95
4.14	Conocimiento adquirido por el nodo 1 después de su primer intercambio.	95
4.15	Total de conocimiento adquirido por el nodo 1.	96

Índice de tablas

1.1	Número aproximado de sensores requeridos para monitorear cada escenario	6
3.1	Equivalencias entre $\Delta(\text{Disagree})$ y $\tau(\text{CAG})$	59
3.2	El tiempo de vida de los nodos	70
4.1	Conocimiento adquirido por el nodo U_1	84
4.2	Conocimiento adquirido por los nodos móviles U_1, U_2 y U_3	85
4.3	Clasificación del nivel de iluminación	94

Capítulo 1

Introducción

1.1 Antecedentes

Los avances tecnológicos de la última década han impulsado el uso de las redes inalámbricas de sensores (WSN¹), las cuales están formadas por un conjunto de nodos sensores que se colocan en alguna región de interés y que, al trabajar en forma colaborativa, obtienen información del entorno y la comparten con otros dispositivos, redes o sistemas informáticos.

El funcionamiento básico de este tipo de redes consiste en que cada sensor adquiere información acerca de un fenómeno físico a la distancia que le permite su *alcance de monitoreo*, y las lecturas obtenidas son enviadas, a través de comunicaciones inalámbricas, ya sea a un nodo principal y remoto llamado *nodo sink*, o únicamente a sus nodos cercanos. El capítulo 2 describe con mayor detalle el funcionamiento de las redes inalámbricas de sensores.

Una de las actividades fundamentales en este tipo de redes es la recolección de datos. Después de que los sensores han sido colocados sobre el escenario que se requiere monitorear, es necesario que de forma sistemática se recolecten los datos adquiridos y sean transmitidos hacia un destino para un futuro procesamiento y análisis. Esta actividad de recolección de datos se enfrenta a las siguientes limitantes:

¹Wireless Sensor Network

- **Energía.** La energía disponible en los nodos sensores se encuentra limitada ya que ésta se suministra mediante baterías. La limitación energética reduce el tiempo de operación de los sensores, así como su alcance de comunicación [1, 2]. Por ejemplo, una típica batería alcalina proporciona cerca de 50 horas de energía a un sensor [1], esto podría traducirse a menos de un mes de operación continua de cada nodo. En una red de grandes dimensiones y de difícil acceso, se vuelve complejo reemplazar o recargar las baterías de miles de nodos sensores estáticos. En caso de que los nodos sean móviles, éstos pueden acercarse a una estación centralizada para recargar sus baterías o incluso, en el caso de teléfonos celulares, son los usuarios los encargados de recargar diariamente la batería. Como ya se mencionó, cada nodo sensor monitorea el fenómeno de interés, realiza algún proceso o cálculo con el dato adquirido y lo transmite hacia algún destino, por lo cual el gasto de energía en cada sensor es ocasionado por tres actividades [3]: monitoreo, procesamiento de datos y comunicación. De estas actividades, la comunicación, que incluye el envío y recepción de datos, es en la que el nodo gasta más energía, por ello y con el objetivo de minimizar dicho gasto, los radios de los sensores son de baja potencia. Esto ocasiona que el rango de comunicación sea relativamente corto y en caso de que éstos envíen sus lecturas hacia el nodo *sink*, deben hacerlo a través de una red ad hoc multisalto. Esto implica que los nodos ubicados entre el origen y el destino deben reenviar los paquetes de datos, y cada retransmisión de información genera un gasto adicional de energía.
- **Cobertura.** En las redes integradas exclusivamente por sensores estáticos, la cobertura está determinada por la configuración inicial pues los sensores permanecen en el mismo lugar en el que fueron instalados. Dicha instalación puede ser predeterminada o aleatoria. Si es predeterminada se puede garantizar cierto nivel de cobertura [4, 5, 6], pero hay aplicaciones en las cuales los sensores no pueden ser instalados manualmente [7] y se recurre a la instalación aleatoria. La estrategia con la que se implementen o instalen los nodos sensores juega un papel importante pues va a determinar qué tan bien está monitoreado cada punto en el área de interés [8, 5]. Sin embargo, debido a la limitante de recursos y a que ciertos escenarios presentan condiciones hostiles, no es trivial diseñar una estrategia que minimice los costos y el número de transmisiones que proporcione una buena cobertura y que mantenga conectada la red [9, 10]. La integración de sensores móviles permite

el monitoreo de áreas más grandes [11, 12] y la cantidad de nodos móviles que se requieren para cubrir determinada área es menor en comparación con la cantidad de nodos estáticos, sin embargo la movilidad en los nodos trae nuevos retos, tales como la posibilidad de cambios frecuentes en la topología de la red, mayores retardos en la recolección de datos e indisponibilidad de información en ciertas zonas.

- **Conectividad.** Es importante que la red se mantenga conectada para que los datos adquiridos por los nodos sensores puedan ser reenviados hacia el nodo *sink* o compartidos con otros sensores vecinos. La densidad de la red [13] está altamente relacionada con el tema de la conectividad, pues debe haber un número suficiente de nodos para cubrir el área a monitorear y que éstos se mantengan conectados pero no debe haber demasiados nodos pues se incrementaría el número de colisiones, ocasionando un deficiente desempeño en las comunicaciones a través de la red.

De la misma forma no debe haber demasiados nodos porque esto puede provocar un incremento en el gasto de energía [14]. Cuando se utiliza una red integrada sólo por sensores estáticos y la instalación se realiza de forma manual y predeterminada, los problemas de conectividad ocurren cuando a los sensores se les agota la batería o presentan alguna falla. En caso de que la red esté integrada sólo por sensores móviles, los problemas de conectividad pueden disminuir o incrementarse dependiendo de la arquitectura que se esté utilizando, así como del patrón de movilidad que siguen los nodos. Por ejemplo, en la figura 1.1(a) se ilustra una red integrada por sensores móviles en la que se utiliza la infraestructura de la telefonía celular y puntos de acceso WiFi para enviar los datos hacia el nodo *sink*. En este caso, es suficiente que los sensores estén bajo la cobertura de alguna antena celular o de algún punto de acceso para que la conectividad se mantenga. En este escenario la conectividad es buena pero el costo de implementar y acceder a la infraestructura es muy alto [15] y en algunos países ni siquiera existe tal infraestructura [16].

En la figura 1.1(b) también se muestra una arquitectura móvil, pero sólo se utilizan puntos de acceso WiFi para enviar los datos hacia el nodo *sink*. En algunas ciudades la cobertura WiFi ya está disponible y es gratuita, pero en algunas otras no hay cobertura completa o el acceso al servicio es costoso. En este ejemplo, la conectividad se mantendrá mientras los nodos sensores tengan acceso a algún servicio

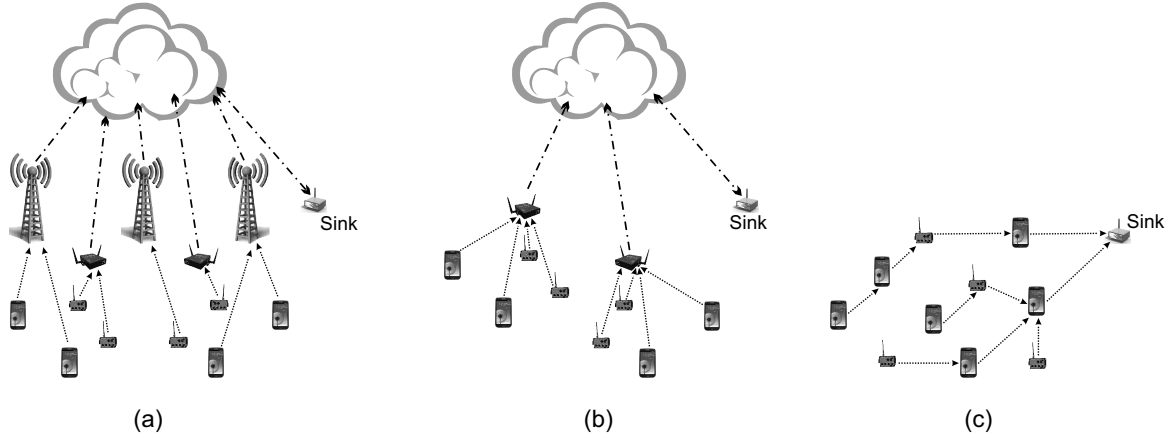


Figura 1.1: (a)Arquitectura móvil donde los sensores utilizan la infraestructura de la telefonía celular y puntos de acceso WiFi para enviar sus datos. (b)Arquitectura móvil donde los sensores utilizan puntos de acceso WiFi para enviar sus datos. (c)Arquitectura móvil donde los sensores no utilizan ningún tipo de infraestructura y se organizan en modo ad hoc para enviar sus datos.

WiFi. Existen algunas soluciones en las que, adicionalmente a los servicios WiFi, se utiliza una red ah hoc multisalto para hacer llegar los datos al nodo *sink*. En la figura 1.1(c) se ilustra una arquitectura móvil que no requiere de la implementación ni el acceso a infraestructura costosa pues sólo se utiliza comunicación ad hoc. Este escenario ilustra una solución económica, sin embargo la movilidad puede afectar la conectividad pues el constante movimiento de los nodos puede ocasionar que, por periodos cortos o largos, algunos nodos queden desconectados y no puedan realizar la retransmisión de la información recibida, ocasionando retardos en la entrega de los datos al *sink*.

- **Escalabilidad.** La escalabilidad es un factor crítico [17] que debe garantizar que aun cuando el área a monitorear crezca y se incremente el número de nodos, el desempeño de la red no debe degradarse significativamente [18, 19, 20]. Los resultados en [18] sugieren que en general las soluciones inalámbricas que utilizan nodos sensores estáticos son inherentemente no escalables cuando el tamaño de la red crece de manera considerable. Algunos trabajos mencionan que las soluciones con sensores móviles son más escalables en comparación con las que sólo utilizan sensores estáticos y se pueden implementar en áreas más extensas [18, 21].

Conforme el área a monitorear es más grande, el número de nodos sensores estáticos requeridos debe incrementarse para mantener cierto nivel de cobertura y así poder obtener una vista completa de lo que sucede con el fenómeno de interés. Por ejemplo, supongamos que el radio de monitoreo de cada nodo es de 6 metros, que se utilizará una cobertura tipo manta, que se solicitará periódicamente una vista completa de la red y que los sensores serán instalados de manera predeterminada utilizando una distribución uniforme (*hexagonal grid* [4]) como la que se muestra en la figura 1.2. Tomando en cuenta todas estas condiciones, se tiene que el número de sensores requeridos crece rápidamente conforme se incrementa el tamaño del área a monitorear, lo cual se ilustra en la figura 1.3.

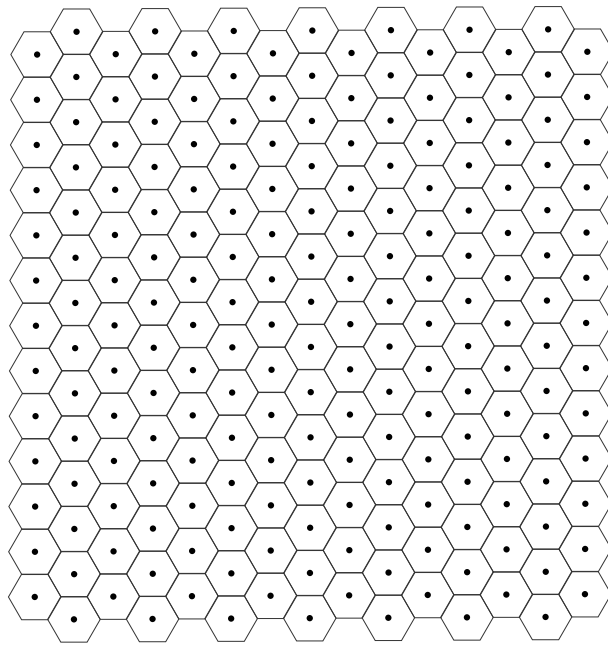


Figura 1.2: *Hexagonal grid*

Si el escenario es un viñedo de $40,000m^2$ el número aproximado de sensores que se requiere es de 241, pero si lo que se desea monitorear es una ciudad de $1,495,000,000m^2$, como el Distrito Federal, entonces se requieren 8,991,028 sensores tal como se muestra en la Tabla 1.1 y se ilustra también en la figura 1.3. Las actividades de mantenimiento y reubicación de dispositivos [22] en una red de sensores de tal magnitud se vuelve inviable, y con frecuencia habrá problemas de conectividad debido a que a los sensores se les agotará la batería.

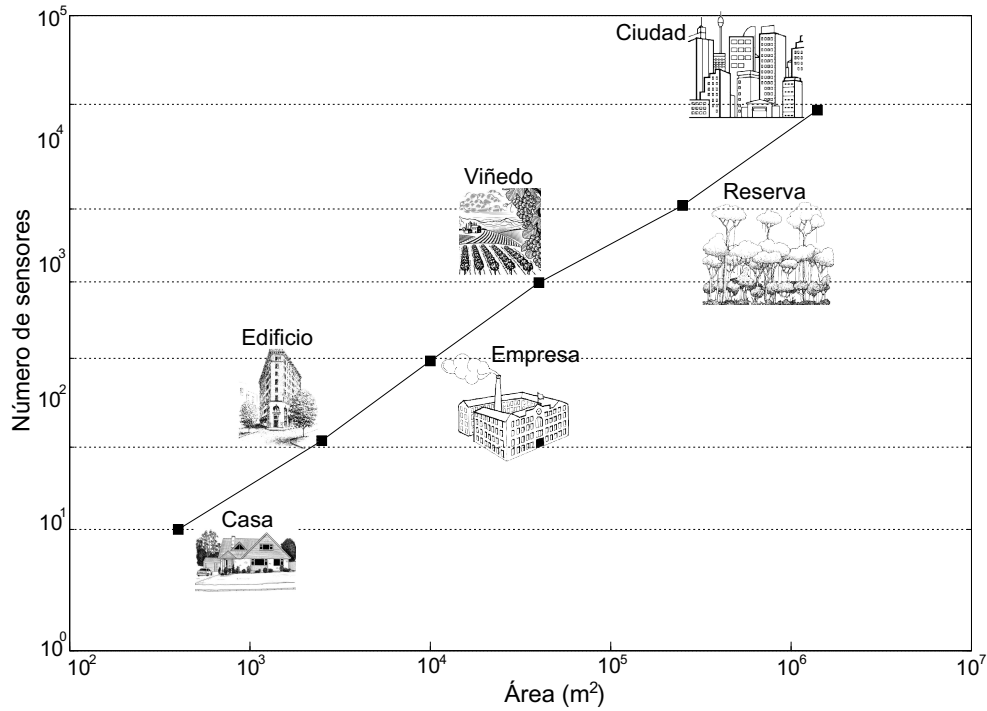


Figura 1.3: Escalabilidad

Escenario	m^2	Número de sensores
Casa	400	2
Edificio	2500	15
Empresa	10000	60
Viñedo	40000	241
Reserva ecológica	250000	1504
Ciudad	149500000	8991028

Tabla 1.1: Número aproximado de sensores requeridos para monitorear cada escenario

Pero ¿qué sucedería si el número de sensores estáticos no creciera tan rápidamente conforme crece el tamaño del área a monitorear? Para ilustrar este supuesto nos apoyaremos en la figura 1.4 con el objetivo de representar el fenómeno de interés que se requiere monitorear, en dicha figura los distintos tonos de gris representan los diferentes valores del fenómeno.

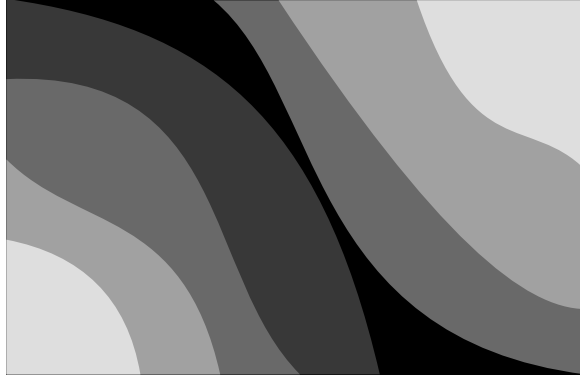


Figura 1.4: Imagen que representa el fenómeno a monitorear.

En las figuras 1.5(a), (b) y (c) se ilustra el escenario de la figura 1.4 monitoreado con 4, 9 y 12 sensores estáticos respectivamente. En caso de solicitar una vista completa en cada una de estas configuraciones, se obtendría lo ilustrado en las figuras 1.5(d), (e) y (f). Al comparar estas últimas tres figuras se puede observar que entre más sensores hay, mayor es la cantidad de información que se obtiene y esto permite tener un panorama más detallado de lo que sucede con el fenómeno monitoreado. Entre mayor sea la densidad de la red, mayor será el detalle de la vista completa. Si la densidad de la red no crece conforme se incrementa el tamaño del área a monitorear, el detalle de la vista completa se verá disminuido.

Se ha demostrado que la integración de sensores móviles para el monitoreo de áreas grandes mejora la cobertura de la red [11, 12], y la cantidad de nodos móviles que se requieren para cubrir determinada área es menor en comparación con la cantidad de nodos estáticos. Por ejemplo, supongamos que en el escenario ilustrado en la figura 1.4 colocamos sensores móviles que se mueven con base en rutas predefinidas como las que se ilustran en la figura 1.6(a). Comparando la figura 1.5(c) con la figura 1.6(a), se observa que con un menor número de sensores móviles se puede cubrir toda el área a monitorear. Incluir dispositivos móviles en una red inalámbrica de sensores mejora la cobertura, permite la escalabilidad y el problema de la limitante de energía se reduce porque, por ejemplo, en caso de que el dispositivo móvil sea un teléfono celular, cada usuario de manera habitual se encarga de recargar la batería y en caso de que los sensores se encuentren instalados en vehículos pues éstos son alimentados por la batería principal del vehículo. Sin embargo, la movilidad trae consigo retos tales como la caducidad de la información, la latencia, la velocidad, etc., los cuales son revisados a continuación:

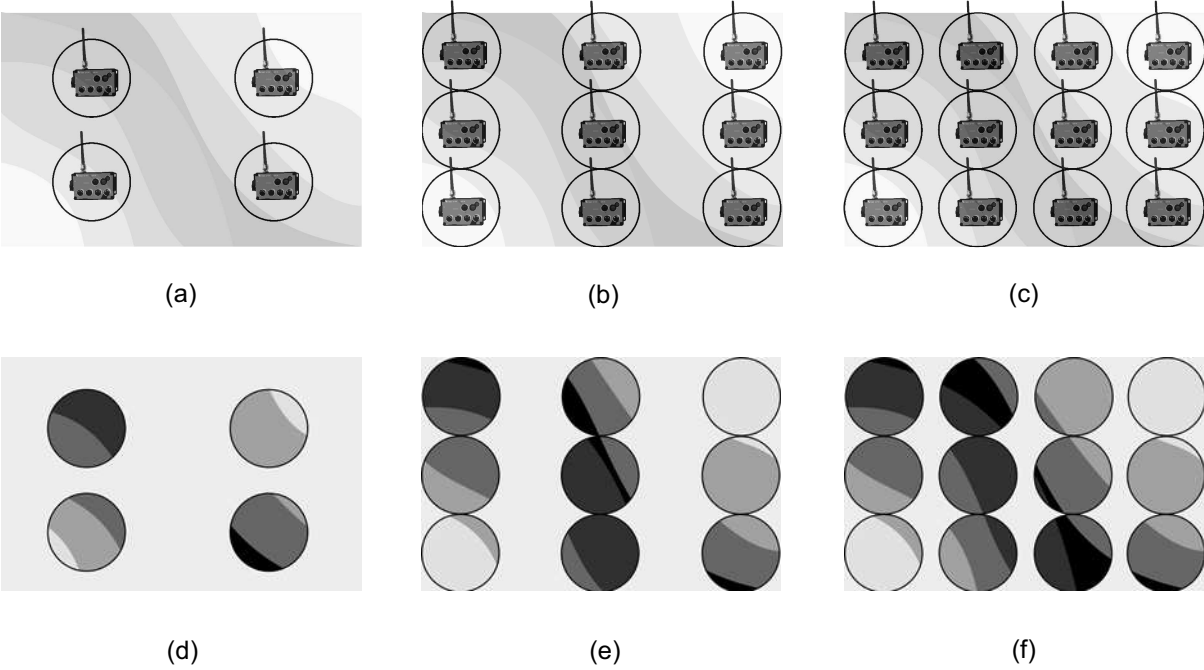


Figura 1.5: (a)(b) y (c) Escenarios monitoreados por 4, 9 y 12 sensores estáticos. (d)(e) y (f) Resultados de solicitar una vista completa de los escenarios (a) (b) y (c) respectivamente.

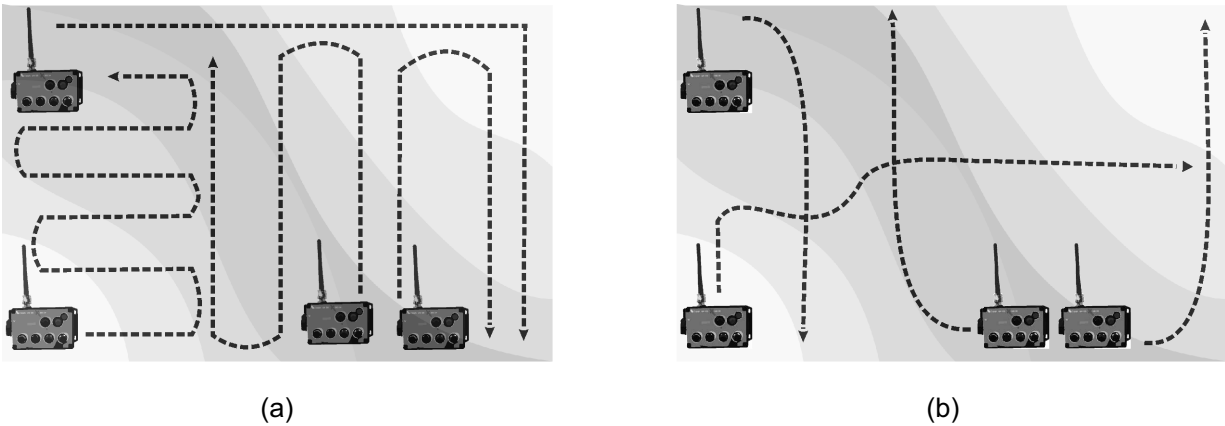


Figura 1.6: (a) Los sensores se mueven siguiendo rutas predeterminadas. (b) Los sensores se mueven con base en rutas no predeterminadas.

- Movilidad y rutas.** Los sensores pueden moverse siguiendo dos tipos de rutas: *predeterminadas* y *no predeterminadas*. Si se utilizan rutas predeterminadas se puede garantizar la cobertura completa del escenario que se quiere monitorear. Pero si

los sensores se mueven con base en rutas no predeterminadas, no se garantiza una cobertura total aun cuando exista una buena densidad de la red pues no se tiene control sobre las rutas. Como ejemplo de sensores siguiendo rutas predeterminadas, se podrían utilizar robots con sensores para monitorear periódicamente algún escenario [23, 3] o se podrían colocar sensores dentro de los vehículos de transporte público que diariamente circulan siguiendo rutas preestablecidas [24, 25]. Como ejemplo de sensores siguiendo rutas no predeterminadas, se podrían instalar sensores dentro de vehículos tipo taxi [26] o incluso utilizar teléfonos celulares [27] para monitorear cierto fenómeno de interés. En la figura 1.6(a) se ilustra un conjunto de sensores que se mueven con base en rutas predeterminadas y se puede apreciar que los sensores cubren totalmente el escenario. Por otro lado, en la figura 1.6(b) se muestra un ejemplo en el que los sensores móviles siguen rutas no predefinidas y se aprecia que la cobertura que logran no es completa.

- **Movilidad y tiempo.** Cuando sólo se utilizan sensores estáticos para monitorear el fenómeno de interés, es posible obtener en cada instante de tiempo t_i información actualizada de cada región monitoreada, sin embargo esto se dificulta cuando sólo se utilizan sensores móviles.

Por ejemplo, supongamos que el escenario ilustrado en la figura 1.4 se divide en 12 zonas, en cada zona se instala un sensor estático como se muestra en la figura 1.7(a) y cada sensor monitorea el fenómeno en cada tiempo t_i , almacenando y enviando la información al nodo *sink*. Con base en este escenario, supongamos que en el tiempo t_2 se solicita a todos los sensores una vista completa de lo que sucede con el fenómeno, por lo que cada sensor enviará las lecturas tomadas en el tiempo t_2 hacia el nodo *sink*. Después de recibir los datos, el nodo *sink* tendrá información actual de todas las zonas. Por otro lado, utilizando el mismo escenario dividido en 12 zonas, ahora colocamos sólo 3 sensores móviles como se muestra en la figura 1.7(b). También en el tiempo t_2 se les solicita a los nodos una vista completa de la red, por lo que cada sensor enviará las lecturas tomadas en el periodo $t_0 - t_2$ al nodo *sink*. Después de recibir los datos el *sink* tendrá datos actuales (t_2) y anteriores (t_0, t_1). Los datos actuales son de la zona en la que se encontraban los sensores en el momento en el que recibieron la solicitud y los anteriores son los datos de las zonas por las que pasaron. Para este caso no se puede tener información actual de todas las zonas para cada tiempo t_i . Adicionalmente, si el sensor aún no ha pasado por determinada zona, tampoco se tendrá información de esa área.

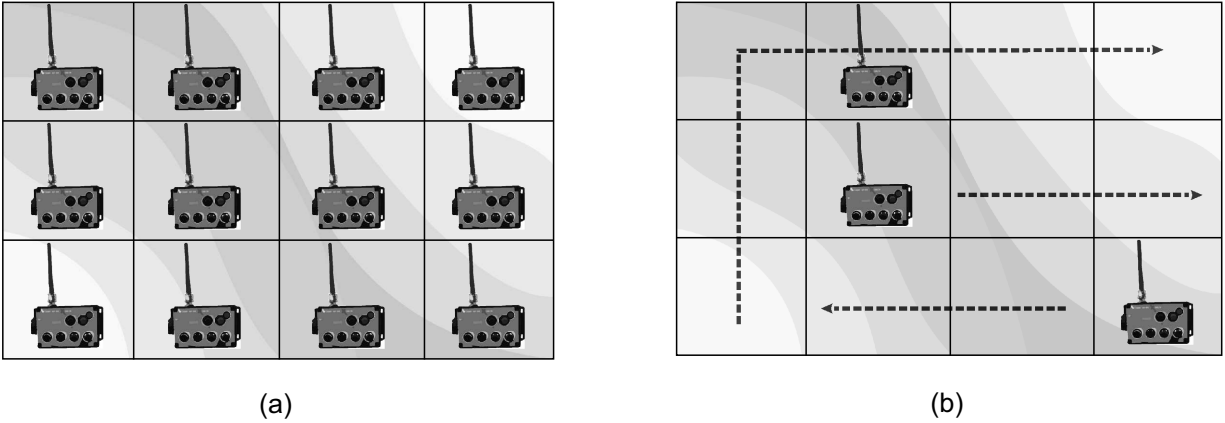


Figura 1.7: (a) Sensores estáticos con información actualizada para cada tiempo t_i . (b) Sensores móviles que sólo tienen información actualizada de determinadas zonas en el tiempo t_i .

- **Movilidad, latencia y obsolescencia de información.** Cuando los sensores estáticos o móviles utilizan infraestructura celular y/o WiFi, el envío y recepción de información es casi inmediato. De esta manera, en caso de que se requiera una vista completa del fenómeno monitoreado los datos obtenidos serán actuales (salvo la situación explicada en el punto anterior), pues el proceso de envío - recepción sólo presentará un retardo vinculado a la calidad de servicio de la infraestructura que se esté utilizando. En caso de utilizar una arquitectura ad hoc sin infraestructura, la información tendrá una latencia variable pues tiene que recorrer alguna ruta multisalto para llegar hasta el nodo *sink*. A esto hay que sumarle que quizá la información de ciertas áreas se tomó en un periodo anterior.
- **Movilidad, infraestructura y convergencia de información.** Los sensores móviles y la infraestructura utilizada pueden beneficiar o dificultar la recopilación de la información para obtener una vista completa de lo que sucede con el fenómeno de interés. Por ejemplo, supongamos dos casos, en el primero se tiene una arquitectura como la que se ilustra en la figura 1.8(a) integrada con sensores móviles, que utilizan la infraestructura celular y puntos de acceso WiFi para enviar sus datos al nodo *sink*. En el segundo caso, se utiliza una arquitectura similar pero no hay nodo *sink*. En su lugar existe una base de datos que será alimentada con la información que envíen cada uno de los nodos móviles como se muestra en la figura 1.8(b). En ambos escenarios aunque un nodo se aleje demasiado y quede fuera de la cobertura

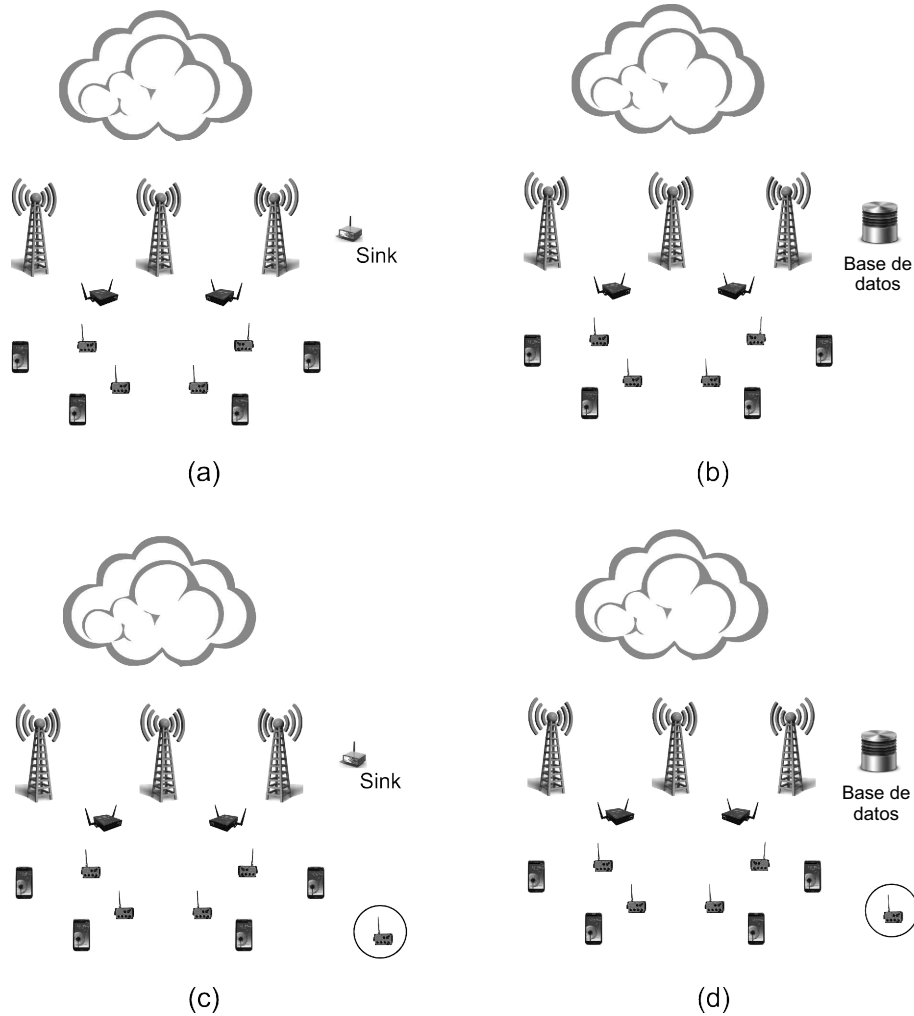


Figura 1.8: (a)Arquitectura móvil que utiliza infraestructura para enviar datos hacia el nodo *sink*. (b)Arquitectura móvil que utiliza infraestructura para enviar y consultar información de una base de datos centralizada. (c)-(d)Después de los movimientos, un nodo queda fuera de la cobertura de la telefonía celular y los puntos de acceso WiFi.

de ambos servicios (celular y WiFi), como se ilustra en las figuras 1.8(c) y (d), los demás nodos podrán seguir enviando sus lecturas. En ambos casos resulta fácil y rápido que la información proveniente de todos los nodos se concentre ya sea en el nodo *sink*, obteniendo así la vista completa del fenómeno, o llegue a la base de datos centralizada para que cualquier otro nodo pueda consultar la información de la vista completa.

Si se quiere evitar el uso de costosa infraestructura se tienen dos casos, el primero se ilustra en la figura 1.9(a), donde hay 6 sensores móviles (identificados con las letras A,B,C,D,E y F), el nodo *sink* es estático y se organizan en una red ad hoc por lo que cada envío de información hacia el *sink* debe seguir una ruta multisalto. En la figura 1.9(a) se muestra un ejemplo en el cual el nodo A envía sus datos hacia el *sink*. En determinado momento los nodos cambiarán de lugar y deberán reorganizarse para no perder la conectividad, como se muestra en la figura 1.9(b) en donde el nodo A envía nuevamente datos hacia el nodo *sink* utilizando una nueva ruta multisalto. Los nodos se mueven una vez más y en la figura 1.9(c) se ilustra que después de dicho movimiento dos nodos quedan desconectados de la red. Debido a movimientos continuos, es posible que la red quede desconectada del nodo *sink* o se desconecten sólo algunos nodos, ambas situaciones ocasionan que la cantidad de información que recibe el nodo *sink* disminuya y se dificulte obtener una vista completa de lo que sucede con el fenómeno monitoreado. El segundo caso utiliza una arquitectura móvil más flexible y es la que muestra en la figura 1.10(a), en donde ya no existe el nodo *sink* y para obtener una vista completa, la información se comparte exclusivamente con los nodos vecinos utilizando comunicación directa. En este ejemplo el nodo A envía datos a los nodos B y C, y el nodo D envía datos a los nodos E y F. Transcurrido determinado tiempo los nodos se mueven a otro lugar, como se ilustra en la figura 1.10(b), y pueden seguir compartiendo información con sus actuales vecinos sin que se requiera que todos los nodos se mantengan conectados y que éstos se reorganicen después de cada movimiento.

- **Movilidad y transmisión de datos.** La movilidad de los nodos puede degradar severamente el caudal de datos (*throughput*) de la red [28]. Esto se debe a que los movimientos pueden causar fallas y cambios en las rutas y por ende la pérdida de paquetes y ciertos retardos. El protocolo de transporte puede interpretar esto como una intensa congestión en la red y entonces activar algún mecanismo para el control de dicha congestión, lo cual ocasionará retransmisiones de los datos y afectación del desempeño [29].

En el capítulo 2 se detalla cómo cada una de éstas limitantes afecta el proceso de recolección de datos.

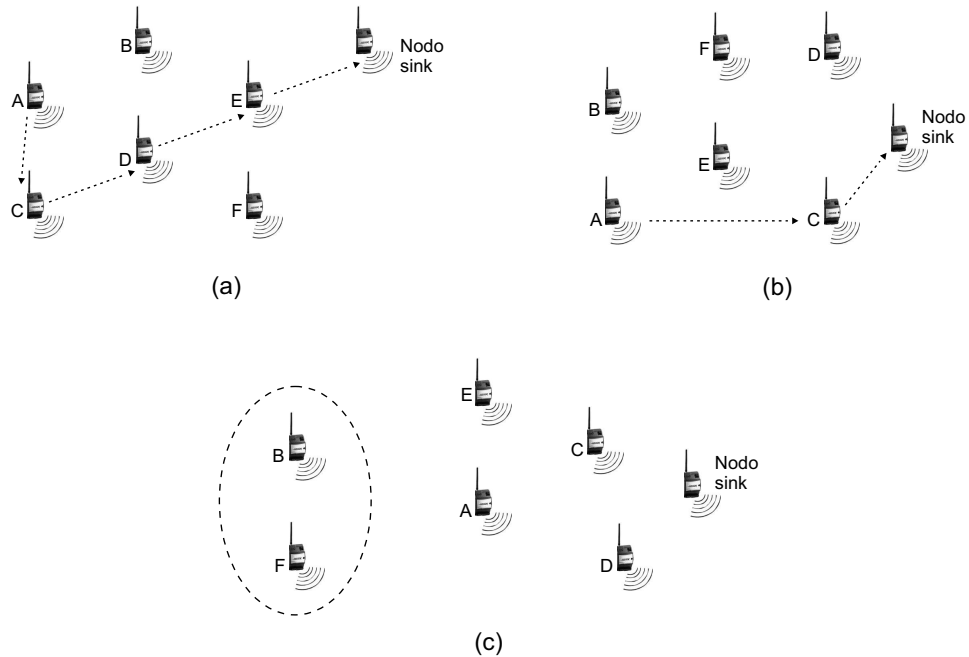


Figura 1.9: (a) El nodo A envía datos al nodo *sink* a través de una ruta multisalto. (b) Después de reorganizarse, el nodo A nuevamente envía datos al nodo *sink* a través de una nueva ruta. (c) Los nodos se mueven y se reorganizan nuevamente pero dos nodos quedan desconectados del resto.

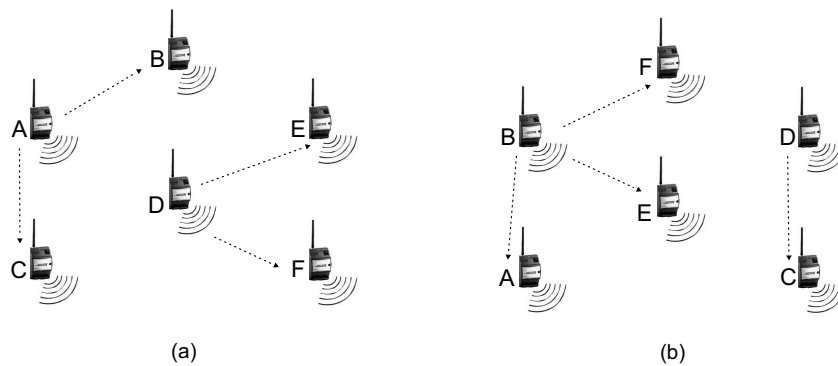


Figura 1.10: (a) Ejemplo de red donde los nodos sólo comparten información con los vecinos que se encuentran a un salto. (b) Después de que se mueven a otro lugar, los nodos pueden seguir compartiendo información con sus vecinos actuales.

1.2 Planteamiento del problema

En este trabajo de investigación se parte del supuesto de que es posible definir estrategias aplicables a los métodos de recolección de datos en redes inalámbricas de sensores, que

permitan obtener información completa acerca de lo que ocurre con la región monitoreada independientemente de su tamaño, incluso cuando la red opere bajo condiciones de limitación energética o con la presencia de nodos móviles.

Para el caso de una pequeña red ad hoc integrada sólo con sensores estáticos, en la cual se envía información al nodo *sink*, nos planteamos las siguientes interrogantes: ¿Cómo se puede obtener una vista completa de lo que sucede con el fenómeno de interés sin requerir que todos los nodos transmitan sus lecturas?; en caso de estimar dichas lecturas ¿cómo acotar el nivel de error?

Para el caso de un área grande de monitoreo en donde se utiliza una arquitectura ad hoc móvil y en la cual no hay nodo *sink* nos cuestionamos lo siguiente: ¿Cómo obtener una vista completa de lo que sucede con el fenómeno de interés sin requerir de infraestructura costosa?

En caso de que se quiera una vista completa de la red, algunas soluciones existentes [30, 31, 32] aprovechan la correlación espacial para agregar los datos en el *cluster head*, y en lugar de enviar una respuesta por cada sensor que pertenezca a dicho *cluster*, sólo se envía una respuesta. En esta tesis argumentamos que no es necesario que todos los nodos transmitan sus datos adquiridos pues la correlación espacial puede explotarse también a nivel de sensor y no sólo a nivel de *cluster head*. Debido a que habrá nodos que permanezcan en silencio, es necesario hacer una rotación periódica para distribuir equitativamente el gasto de energía.

Por otro lado, si el área a monitorear es grande, por ejemplo una ciudad, y es muy costoso el acceso a la infraestructura o no se tiene y se requiere obtener una vista completa de lo que sucede con el fenómeno de interés, argumentamos que dicho conocimiento global puede obtenerse utilizando exclusivamente intercambios locales entre nodos sensores móviles cercanos.

1.3 Objetivos

El objetivo general de este trabajo es diseñar métodos eficientes de monitoreo y recolección de datos para aplicaciones en las que se requiera obtener periódicamente una vista completa de lo que sucede con el fenómeno en toda la red.

Para el caso de una red pequeña donde los nodos no son móviles, nuestro objetivo específico es diseñar un método para ahorrar energía durante la recolección de datos y así extender el tiempo de vida de la red.

Para el caso de una red más grande con nodo móviles, en donde se asume que el problema de la energía disminuye gracias a que hay algún factor adicional que constantemente recarga cada nodo sensor, el objetivo es diseñar un método que nos permita conocer la información de toda una área, por ejemplo una ciudad, sin requerir de infraestructura costosa para lograr así un ahorro económico importante.

1.4 Contribución y relevancia de la tesis

La contribución de esta tesis se resume a continuación:

El método *Disagree* [33] es un nuevo método para la adquisición de datos en las redes inalámbricas de sensores estáticos, que explota la correlación espacial de datos al nivel del sensor gracias a que la solicitud de información se ajusta a los valores de cada vecindario (*cluster*). Más que solicitar a todos los sensores el envío de información para obtener una vista completa de la red, *Disagree* solicita información a aquellos sensores cuyas lecturas estén fuera de un intervalo centrado en la lectura reportada por el nodo árbol (*cluster head*) con quien dichos sensores se encuentran asociados. Esto permite que existan nodos sensores que permanecen en silencio con lo cual se logra un importante ahorro de energía. Las lecturas de los nodos silenciosos pueden ser estimadas fácilmente, y el error de estimación puede ser controlado a través de un parámetro. *Disagree* es una solución centralizada que utiliza sensores estáticos y no está orientado a monitorear fenómenos en áreas grandes como por ejemplo una ciudad. Se ha demostrado que la integración de sensores móviles para el monitoreo de este tipo de escenarios mejora la cobertura de

la red [11, 12], y la cantidad de nodos móviles que se requieren para cubrir una ciudad de más de $1,000 \text{ km}^2$ es menor en comparación con la cantidad de nodos estáticos sin embargo la latencia se incrementa. Y esto es el punto de partida para nuestra segunda propuesta llamada *Urbihoc*, el cual es un método que utiliza nodos sensores móviles para monitorear ciertos fenómenos en áreas grandes. *Urbihoc* aprovecha la cobertura que la movilidad proporciona y a partir de intercambios locales entre nodos cercanos, se obtiene una vista completa o conocimiento global del comportamiento de un determinado fenómeno en una ciudad, sin necesidad de implementar o utilizar infraestructura costosa. Los nodos móviles pueden ser teléfonos celulares (*smartphones*), unidades de transporte público, vehículos particulares, etc., estos dos últimos tienen la ventaja de que pueden ser alimentados por la batería del automóvil y se les pueden instalar sensores más especializados. Los teléfonos celulares están más limitados en los sensores que incluyen y en batería, aunque regularmente son recargados por sus respectivos dueños, sin embargo pueden llegar a lugares de difícil acceso para los vehículos. *Urbihoc* no está orientado a algún tipo de nodo móvil en particular.

Actualmente las soluciones construidas para las redes de sensores tienen propósitos muy específicos, por lo que no hay aplicaciones genéricas que trabajen en una gran variedad de escenarios. Este trabajo de investigación es relevante porque el criterio de solicitud de información utilizado por *Disagree* puede servir a otros trabajos de investigación que requieran lograr un ahorro global en el gasto de energía. Por otro lado, la relevancia de *Urbihoc* radica en que los beneficios mostrados en los resultados, representan un factor que puede motivar a los usuarios a compartir su información local recolectada sin utilizar infraestructura, por lo que podría aplicarse en escenarios donde ésta no existe o ha sido inhabilitada por condiciones adversas.

1.5 Estructura de la tesis

La estructura de nuestro estudio es la siguiente:

En el capítulo 2 se presentan algunos conceptos fundamentales y el estado del arte relacionado con la recolección de datos en las redes inalámbricas de sensores. En el capítulo 3 presentamos *Disagree*, un método eficiente para recolectar datos en una red inalámbrica

de sensores estáticos que ayuda a reducir el gasto de energía. En este capítulo se presenta el diseño detallado de *Disagree* y su evaluación utilizando simulaciones, experimentos y una implementación.

En el capítulo 4 introducimos *Urbihoc*, el cual es un método eficiente para recolectar datos en áreas grandes sin necesidad de utilizar infraestructura costosa. *Urbihoc* es aplicable a redes inalámbricas ad hoc con sensores móviles. En este capítulo presentamos las simulaciones, experimentos y la implementación que sirvieron para evaluar la viabilidad de este método.

En el capítulo 5 se presentan las conclusiones de nuestro trabajo y se menciona de forma general las líneas de investigación a seguir y el trabajo futuro. Posteriormente se incluye el anexo A, en donde se listan las publicaciones en las que se participó.

Capítulo 2

Conceptos fundamentales y estado del arte

2.1 Conceptos fundamentales

Un sensor es un dispositivo que mapea una característica física de algún entorno a una medición cuantitativa [1]. Los avances tecnológicos en las comunicaciones inalámbricas y en la electrónica han permitido que los sensores sean cada vez más pequeños, de menor costo y con menor consumo energético. Esto a su vez ha permitido integrar dichos sensores a dispositivos equipados con un microprocesador, memoria y un radio para transmitir vía inalámbrica los datos recolectados [2], o incluso que sean incorporados a dispositivos móviles como son los teléfonos celulares (*smartphones*) o que sean instalados dentro de vehículos particulares o en sistemas de transporte público [34]. La figura 2.1 ilustra algunos dispositivos¹ que incluyen sensores.



Figura 2.1: Tipos de dispositivos con sensores

¹A lo largo de este trabajo nos referiremos a estos dispositivos simplemente como nodos sensores o sensores.

La figura 2.2 muestra los componentes clave de cada uno de los dispositivos ilustrados en la figura 2.1: procesador, memoria, radio(s), sensor(es), fuente de energía y algunos dispositivos pueden incluir GPS².

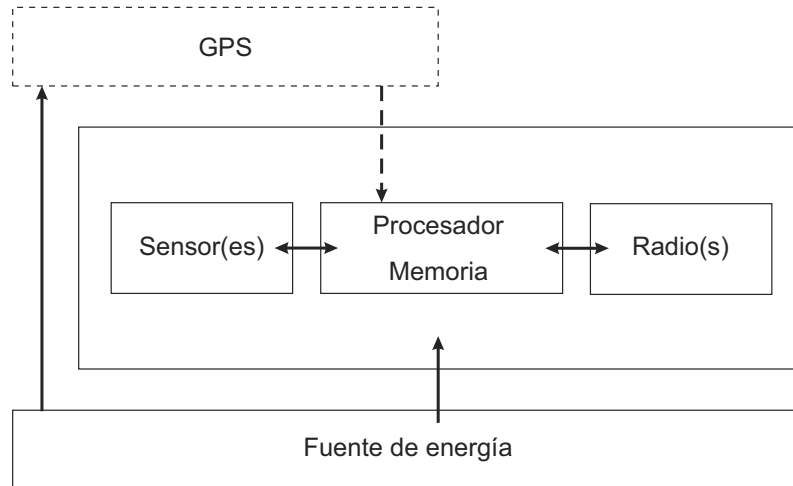


Figura 2.2: Componentes clave de los sensores

Una red inalámbrica de sensores (WSN) puede ser definida como un conjunto de pequeños dispositivos inalámbricos, denominados sensores, que se utilizan para el monitoreo físico de diferentes fenómenos, tales como el sonido, la vibración, el movimiento, los niveles de ciertas sustancias en el ambiente, la detección de fuego, etc. De manera cooperativa, cada uno de estos sensores, transmiten los datos adquiridos a través de la red hacia uno o más destinos en donde dicha información pueda ser analizada y utilizada. Debido a que los nodos tienen energía limitada, estos deben restringir su alcance de transmisión a unos cuantos metros para maximizar el tiempo de vida de la red [35]. Esto ocasiona que la mayoría de las redes WSN deban operar en un esquema multisalto.

2.2 Características de las WSN

Las redes inalámbricas de sensores han permitido la creación de nuevas aplicaciones y requieren del diseño de protocolos y métodos poco convencionales debido a que, en general, los dispositivos de monitoreo están limitados en cuanto a capacidades de procesamiento, memoria y energía, por lo que debe haber un equilibrio adecuado en el uso de estos recursos. Esto ha motivado que desde la última década exista un gran esfuerzo

²Global Positioning System

en actividades de investigación, estandarización e inversiones industriales encaminados a mantener dicho equilibrio [36].

Los sensores usualmente se encuentran dispersos en el escenario a monitorear. Cada uno de estos nodos tiene la capacidad de adquirir datos, como resultado del monitoreo de algún fenómeno de interés, realizar un determinado procesamiento con ellos y enviarlos hacia el nodo *sink* o a sus nodos vecinos. Dependiendo de cómo los sensores se encuentren organizados, los datos pueden ser encaminados hacia el nodo *sink* a través de una red inalámbrica ad hoc multisalto o de una red inalámbrica con infraestructura. En la figura 2.3(a) se ilustra un ejemplo en el cual el sensor identificado con la letra A envía sus lecturas al nodo *sink*. Debido a que entre el nodo A y el nodo *sink* hay una distancia mayor a sus respectivos alcances de transmisión, la comunicación directa no es posible, por ello los nodos intermedios D y E deben recibir y reenviar la información para hacerla llegar a su destino. En este ejemplo, la comunicación hacia el nodo *sink* está utilizando una técnica *multisalto*³[3, 1]. En la figura 2.3(b) se muestra un ejemplo en el que el sensor A comparte sus lecturas sólo con los nodos vecinos. En este escenario, la comunicación ocurre a *un salto*⁴[1] ya que sólo se realiza en forma directa, esto es, sin la intervención de nodos intermedios.

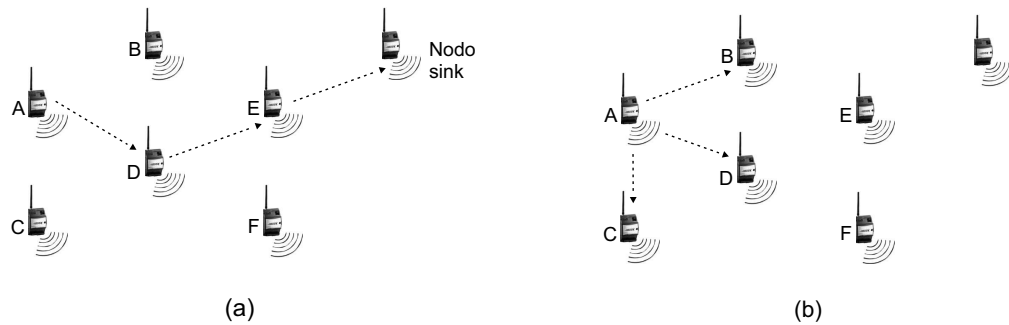


Figura 2.3: (a) El sensor A envía sus lecturas obtenidas al nodo *sink* (transmisión multisalto). (b) El sensor A comparte sus lecturas obtenidas con los nodos cercanos (transmisión de un salto).

Normalmente, los sensores envían sus lecturas en respuesta a las solicitudes que reciben de otros nodos cercanos o del nodo *sink*. Dependiendo de la aplicación en la red

³ *Multihop*

⁴ *Single-hop*

WSN, el modelo de entrega de datos hacia el nodo *sink* puede ser *continuo*, *orientado a eventos* (*event-driven*), *orientado a solicitudes* (*query-driven*) e *híbrido*. En el modelo continuo cada sensor envía periódicamente sus lecturas hacia el nodo *sink* u otros nodos, la frecuencia de los envíos depende de las necesidades de la aplicación [37, 3]. Con este modelo se puede obtener con cierta regularidad una *vista completa* de lo que sucede con el fenómeno de interés pero, en general, este modelo implica un mayor consumo energético. En el modelo orientado a eventos la transmisión de los datos es iniciada cuando se detecta determinado evento o valor, en general se espera que en este tipo de redes los nodos permanezcan inactivos la mayor parte del tiempo [1, 3]. En el modelo orientado a solicitudes, los nodos reciben las solicitudes de información y sólo si los datos que contienen satisfacen dicha demanda inician el envío de datos [37]. A su vez, este modelo puede estar centrado en la ubicación (*location-centric*) o centrado en los datos (*data-centric*). Por ejemplo, las solicitudes centradas en la ubicación pueden ser dirigidas a toda la red o enviadas a un área en específico. Una solicitud centrada en los datos consistiría en requerir a todos los sensores el envío de su información, siempre y cuando ésta cumpla con ciertas condiciones. Algunas aplicaciones [1] utilizan un modelo híbrido, es decir, combinan dos o más de los modelos anteriores y así por ejemplo, los sensores podrían enviar reportes periódicos al nodo *sink*, y también podrían transmitir su información cuando recibieran solicitudes de sus nodos vecinos o del propio *sink*. Otro ejemplo de modelo híbrido es el *orientado a eventos solicitados* (*event-driven query*), en donde se les puede solicitar a los sensores que en lugar de monitorear el evento e_1 (por ejemplo, la temperatura) ahora realicen el monitoreo del evento e_2 (por ejemplo, la humedad).

Los trabajos de investigación en las redes WSN están orientados a resolver un problema en específico, por lo que la arquitectura necesaria depende de las metas de la aplicación. Sin embargo, hay una serie de aspectos que las soluciones que incluyan redes WSN deben considerar tales como la implementación, la infraestructura, la topología, los protocolos de comunicación y los tipos de cobertura, cada uno de los cuales son explicados a continuación:

2.2.1 Implementación e infraestructura

Existen varias estrategias para implementar una red inalámbrica de sensores [23], los sensores pueden estar incrustados en el asfalto de las carreteras y en muros de edificios

o instalados en árboles, en autos, en teléfonos celulares, etc. Estos sensores pueden ser colocados de manera determinística por humanos o por robots o instalados de manera aleatoria. Una vez instalados, los nodos sensores deben auto-organizarse para poder comunicarse. Existen diferentes modalidades de comunicación que pueden ser utilizadas en una red WSN, las más conocidas son las redes con infraestructura y las redes ad hoc. En las redes con infraestructura, los nodos sólo pueden comunicarse directamente con los puntos de acceso (*Access point*) o estaciones base y cualquier comunicación entre nodos debe realizarse a través de un punto de acceso. El número de puntos de acceso dependerá del alcance de comunicación y el área que debe ser cubierta por los nodos sensores. En las redes ad hoc, los nodos pueden comunicarse con otros nodos sin requerir de infraestructura, aquí los nodos pueden actuar como encaminadores y reenviar mensajes utilizando rutas multisalto, en particular entre los nodos que se encuentren separados del nodo *sink* a distancias mayores que sus alcances de transmisión.

2.2.2 Topología

En su forma más simple, la disposición física de una red WSN forma una estrella en donde cada sensor es capaz de conectarse directamente con el nodo *sink* o con un punto de acceso. Sin embargo, en las redes WSN con un gran número de nodos la topología normalmente se construye con base en un grafo arbitrario y, con frecuencia, dicho grafo está basado en un árbol. La topología afecta a la red en cuestiones de retardo, robustez y capacidad de transmisión. La complejidad del encaminamiento de los datos y su procesamiento dependerá de la topología, la cual también determinará cuáles serán las actividades que deben realizar los nodos dependiendo del lugar que ocupen en la estructura formada. A continuación se mencionan las formas de organización [38] más utilizadas:

- **Plana.** En las redes planas [3, 39, 40] todos los nodos sensores realizan funciones propias del encaminamiento, por lo que participan en el reenvío de información cuando cada sensor transmite individualmente sus lecturas hacia el nodo *sink* a través de una ruta multisalto.
- **Jerárquica.** En este tipo de redes [30, 3] los nodos se organizan en grupos llamados *clusters*, y en cada *cluster* hay un nodo llamado *cluster head* o cabeza de grupo; cada sensor se asocia a un *cluster head* en específico. Aquí los nodos tipo *cluster*

head tendrán actividades adicionales como, por ejemplo, recibir los datos de los nodos que se encuentran asociados con él, agregarlos en caso de que se requiera y reenviarlos al siguiente *cluster head* o al *sink* [41, 42, 43].

- **Conjunto dominante conectado.** En este tipo de redes los sensores se organizan utilizando un conjunto dominante conectado (CDS⁵) para formar la estructura central (*backbone*). Los nodos que no son parte del conjunto dominante son adyacentes al menos a un nodo que pertenece a dicho conjunto. En algunas soluciones basadas en este esquema, los miembros del conjunto dominante son *cluster heads*, los cuales también agregan las lecturas que les envían los demás sensores y las encaminan hacia el *sink*.

2.2.3 Protocolos de comunicación

Los protocolos de comunicación utilizados en las redes WSN se caracterizan porque procuran en su funcionamiento el ahorro de energía, la coordinación de tareas entre sensores y la gestión de la movilidad [44, 3, 45]. Este conjunto de protocolos está integrado por las capas física, enlace, red, transporte y aplicación, como se observa en la figura 2.4.

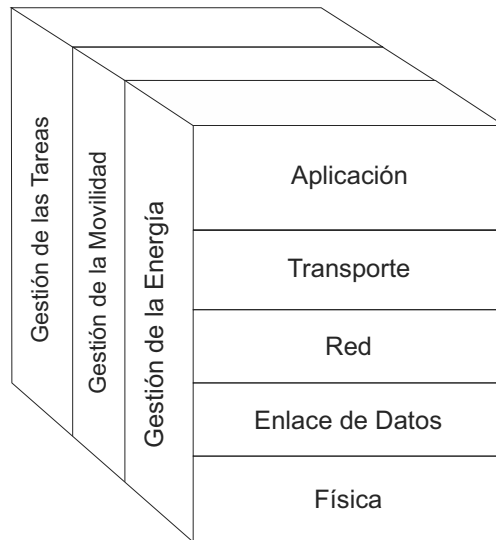


Figura 2.4: Protocolos de comunicación utilizados por las redes WSN.

⁵Connected Dominating Set

Ha habido varios esfuerzos para construir protocolos de comunicación poco convencionales para las redes WSN, a continuación revisaremos algunos de ellos. La *capa física* se encarga de las operaciones de la interfaz del radio como son la selección de la frecuencia, la potencia de transmisión, la modulación, la codificación y detección de las señales. La *capa de enlace* es responsable de controlar el acceso al medio entre los diferentes competidores, mantener la conectividad entre los nodos y proporcionar oportunidades equitativas de transmisión. Los protocolos de acceso al medio basados en el esquema CSMA/CA⁶, que se utilizan en diversos tipos de redes inalámbricas como las del estándar IEEE⁷ 802.11, no son adecuados para las redes WSN porque los sensores invierten mucha energía en la detección de las transmisiones activas. Uno de los primeros protocolos eficientes para esta capa es SMAC⁸ [46], en el cual los nodos siguen un ciclo de escuchar/dormir, es decir, los nodos tienen periodos de sueño que son más largos que los periodos en los que están escuchando el medio. A partir de esta propuesta que, ya lograba un ahorro importante de energía, surgieron nuevas mejoras como [47, 48, 49, 50, 51, 52, 53] las cuales también están enfocadas en proporcionar protocolos para la capa de enlace que logren un ahorro de energía.

La *capa de red* se encarga del encaminamiento de los paquetes hacia el destino. Debido a que las redes WSN tienen una naturaleza de aplicación específica, las técnicas de encaminamiento que utilizan están centradas en los datos, es decir, interesa más saber qué valores se adquirieron en determinado lugar a qué direcciones tienen determinados nodos [35]. Existe una gran variedad de protocolos de encaminamiento para las redes WSN, los cuales se clasifican con base en diferentes criterios por ejemplo, dependiendo de cuál es la estructura de la red pueden ser planos [54, 55, 56, 57, 58, 59], jerárquicos [30, 60, 61] y basados en la ubicación [62]. También pueden ser clasificados dependiendo del encaminamiento que utilice el protocolo en, encaminamiento multiruta, encaminamiento basado en solicitud (*query*) [58, 57], basado en negociación [55], basado en calidad de servicio [63]. También los protocolos pueden ser clasificados de acuerdo a cuál es el destino del paquete [1], si es a un sólo nodo (*unicast*), a un conjunto de nodos (*multicast*) [64] o a todos los nodos de la red (*flooding*) [65]. La mayoría de los trabajos mencionados incluyen en su funcionamiento algún mecanismo para ahorrar energía [66].

⁶Carrier Sense Multiple Access with Collision Avoidance

⁷Institute of Electrical and Electronics Engineers

⁸Sensor Medium Access Control Protocol

Un protocolo frecuentemente utilizado en la *capa de transporte* es el protocolo TCP⁹, el cual se encarga de asegurar la confiabilidad de la transmisión, el control de flujo y la congestión de la red. Debido a la complejidad de las actividades que se realizan en esta capa [67], no es adecuado utilizar TCP (tal cual se usa en las redes en otro tipo de redes) para las redes WSN [45]. Por ello, al igual que en la capa de red, han surgido una serie de trabajos que buscan adaptar este protocolo a las redes WSN como [68, 69, 67, 70, 71, 72]. Por ejemplo, con el protocolo CODA¹⁰ [67] cada sensor monitorea regularmente la ocupación de su buffer, si ésta es alta el sensor escucha las condiciones del canal para confirmar o no la existencia de congestión. Si la congestión es detectada, el sensor envía un mensaje a todos sus vecinos indicándoles que disminuirá sus transmisiones para reducir la congestión. Debido a que cada sensor realiza dicho procedimiento, la congestión disminuye y con ello se logra un ahorro de energía pues se reduce la cantidad de transmisiones.

Dependiendo del fenómeno que se requiera monitorear, diferentes tipos de *software* deben ser construidos y utilizados sobre la *capa de aplicación*. La *gestión de la energía* está relacionada con la administración de los recursos energéticos, por ejemplo, el nodo puede apagar su receptor inmediatamente después de recibir un mensaje de alguno de sus vecinos, con lo cual evitará la llegada de paquetes duplicados. También cuando el nivel de energía de un sensor sea bajo, el nodo sensor podría informar a todos sus vecinos que tiene batería baja y que no podrá participar en el enrutamiento de los próximos mensajes. La *gestión de la movilidad* incluye actividades como que cada sensor detecte y registre el movimiento de los nodos sensores vecinos para así mantener actualizadas las rutas. La *gestión de las tareas de coordinación* equilibra y organiza el monitoreo en una región, de tal manera que no todos los sensores de un área en específico son requeridos para monitorear al mismo tiempo, como resultado, algunos nodos sensores pueden realizar la tarea de otros nodos dependiendo de su nivel de energía. Cada una de estas técnicas de gestión se integran con las las capas de comunicación, por ejemplo, la capa de enlace y de transporte pueden implementar métodos para el ahorro de energía, la capa de red puede gestionar la movilidad, etc. Estos elementos de gestión son necesarios pues permiten que los sensores puedan trabajar juntos de manera eficiente en cuanto al

⁹Transport Control Protocol

¹⁰Congestion Detection and Avoidance

uso de energía, enrutamiento de los datos entre sensores y compartir recursos entre los nodos. Sin estos elementos de gestión cada sensor trabajaría individualmente, y desde el punto de vista de la red de sensores, es más eficiente si los nodos trabajan de forma colaborativa pues así se prolonga el tiempo de vida de dicha red [44, 3].

2.2.4 Tipos de cobertura

Como ya se mencionó, en algunas aplicaciones lo importante es detectar un evento y reportarlo, sin embargo existen otras aplicaciones [37] en las que es necesario conocer cuál es el comportamiento general del fenómeno de interés en todo un escenario, es decir, se requiere obtener una vista completa de lo que sucede con la variable monitoreada. Por ello, el tipo de cobertura que debe utilizarse depende del tipo de aplicación.

En [8], los autores definen tres tipos de cobertura: *cobertura tipo manta o capa (blanket)* en la cual cada punto dentro del área de interés se encuentra dentro del rango de monitoreo de al menos un sensor. La *cobertura de objetivo (target)*, la cual consiste en monitorear un número fijo de objetivos estáticos o móviles. Al tercer tipo se le denomina *cobertura de barrera (barrier)* y ésta consiste en detectar movimientos que crucen una barrera u obstáculo al cual previamente se le instalaron sensores. Dependiendo de la aplicación es el tipo de cobertura que debe seleccionarse, por ejemplo si se desea obtener periódicamente una vista completa de lo que sucede en toda un área, lo ideal es utilizar la cobertura tipo manta como se ilustra en la figura 2.5(a). La cobertura tipo objetivo puede ser utilizada en aplicaciones de rastreo como se muestra en la figura 2.5(b). El uso del tercer tipo se ilustra en la figura 2.5(c) y se puede aplicar cuando se requiere detectar algún evento relacionado con el control de ingreso a un sitio o la detección de la invasión de cierta zona.

2.2.5 Clasificación de las aplicaciones

Uno de los aspectos motivantes para realizar investigación en las redes WSN es la diversidad de aplicaciones que pueden ser desarrolladas para solucionar diferentes tipos de problemas. A continuación se mencionan varias categorías, incluyendo algunos ejemplo en cada una de éstas, en las que las redes inalámbricas de sensores están siendo utilizadas con éxito:

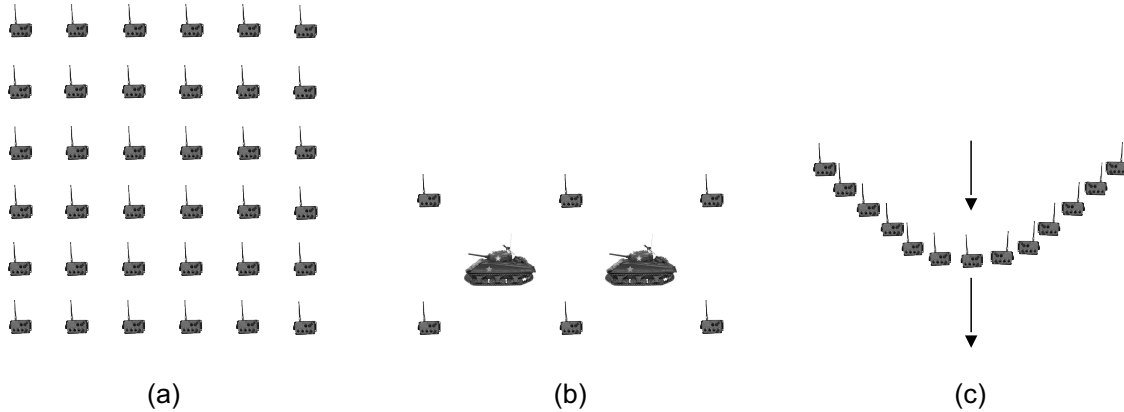


Figura 2.5: (a) Cobertura tipo manta. (b) Cobertura de objetivo. (c) Cobertura de barrera.

- Aplicaciones militares. Para la vigilancia y el reconocimiento de diversos objetos [73].
- Aplicaciones ambientales. Para el monitoreo de cambios en las condiciones ambientales de animales y plantas [74, 75], de la calidad del aire y del agua [76, 77], de los glaciares [78], apoyo a la agricultura [79].
- Aplicaciones de salud. Para el seguimiento y monitoreo de ciertas variables biomédicas [80].
- Espacios y servicios inteligentes. En edificios inteligentes [81, 82], en sistemas de transporte inteligente (ITS¹¹) [83], automatización de casas [84].
- Localización. De personas u objetos [85], exploración científica en lugares peligrosos o de difícil acceso [86].
- En la industria. Identificación de productos [1, 2, 86].
- En la construcción. Para el monitoreo de estructuras [87].

Debido a esta variedad de aplicaciones, el uso de sensores se ha extendido y ahora se han integrado a diversos objetos de uso cotidiano como lo son los vehículos [24, 25] y los teléfonos celulares. En particular estamos interesados en el uso de los *smartphones*,

¹¹Intelligent Transportation Systems

los cuales ya no sólo se utilizan para comunicación de voz, sino que se han convertido en sofisticadas herramientas capaces de ejecutar diversas aplicaciones [88]. Dichos dispositivos incorporan un conjunto de sensores tales como acelerómetro, brújula digital, giroscopio, GPS, micrófono, cámara [27], sensor de luz y sensor de proximidad. En un futuro cercano quizá estos dispositivos tendrán sensores para monitorear fenómenos como la contaminación, humedad, nivel de polen y el grado de alcohol en la sangre. [89, 90], etc.

2.3 Recolección de datos y estado del arte

Las redes WSN involucran varias etapas, la instalación de los sensores, la autoconfiguración de la red, el monitoreo del fenómeno de interés y el envío de la información adquirida hacia el nodo *sink*. Como ya se mencionó, la instalación de los sensores puede ser determinística o aleatoria y dependiendo cómo se auto-configuren los nodos, los protocolos de configuración pueden clasificarse en tres grupos: *centralizados*, *locales* e *híbridos* [35]. Los protocolos *centralizados* requieren que cada sensor tenga información global de la red para que puedan tomar decisiones. Este tipo de protocolos involucran el uso de estructuras topológicas como la del árbol mínimo (MST¹²), cuya estructura completa no puede ser construida a partir de, exclusivamente, información local. Estos protocolos se desempeñan bien cuando las redes WSN son pequeñas. En los protocolos *locales*, como su nombre lo indica, los sensores sólo requieren de conocimiento local para tomar decisiones y una cantidad muy limitada de información adicional, como por ejemplo la ubicación del nodo *sink*. Algunos protocolos locales son *híbridos*, es decir requieren al inicio un pre-procesamiento centralizado tal como la construcción de alguna topología y esta actividad sólo se lleva a cabo un vez. Un ejemplo típico de este tipo es la construcción de una estructura basada en *cluster head*.

Una de las tareas más importantes en las redes WSN es el envío de los datos adquiridos por los sensores hacia algún destino, a esta actividad también se le conoce como *recolección de datos*. La recolección de datos tiene un componente de encaminamiento implícito que puede o no involucrar agregación de información. La gran variedad de requerimientos y objetivos de las diferentes aplicaciones en las redes WSN, determinan los

¹²Minimum Spanning Tree

criterios de diseño, los cuales a su vez están fuertemente relacionados con los siguientes factores [91]:

1. **El número de nodos fuente** ¹³ **y de nodos *sink* en la red.** Con base en este factor, las aplicaciones pueden ser *múltiples fuentes-un sink*, *una fuente-múltiples sink* y *múltiples fuentes-múltiples sink*. Por ejemplo, una aplicación para monitorear el ambiente pertenece a la categoría de *múltiples fuentes-un sink*. Un sistema de reportes que distribuye información acerca de las condiciones del tránsito (tráfico) vehicular a muchos conductores (*sinks*), puede estar en la categoría *una fuente-múltiples sink*.
2. **El equilibrio entre energía, ancho de banda, latencia y precisión de la información.** Normalmente no es posible optimizar el desempeño de todos estos aspectos, por lo que se debe realizar una priorización de requerimientos para así determinar cuáles de estos factores son los más importantes en la aplicación. Por ejemplo, en aplicaciones de misión crítica *end-to-end*, como el reporte de fuego o la existencia de alguna bomba, la latencia es el factor más importante y los otros factores pasan a un nivel secundario, aún a expensas de un consumo adicional de energía [23].

Los métodos de recolección de datos pueden ser clasificados tomando en cuenta distintos aspectos, a continuación se explican brevemente las clasificaciones más relevantes y a la par se mencionan los trabajos más importantes que pertenecen a dichas categorías. También es importante mencionar que el detalle de algunos de estos trabajos se presenta en las secciones del trabajo relacionado de los capítulos 3 y 4.

El proceso de recolección de datos usualmente consiste en dos pasos: solicitud y respuesta, en la solicitud se puede requerir la detección de algún evento, de algún valor que supere cierto umbral, de algún valor que no esté en determinado rango o incluso se le puede indicar al nodo el envío de reportes periódicos. De esta manera un nodo *sink* o un usuario envía una solicitud a una red WSN y los sensores que cumplan con dicha solicitud envían su respuesta al *sink*. Con base en cómo los nodos sensores reporten un evento o envíen información hacia el *sink*, los métodos pueden clasificarse en los siguientes tipos:

¹³Un nodo fuente es cualquier nodo que, como resultado de monitorear el fenómeno de interés, adquiere información que debe ser enviada hacia algún lado.

- **Envío directo por sensores individuales.** Esta es la forma más simple de reportar datos y consiste en que cada nodo, utilizando la potencia de transmisión suficiente, envíe un paquete que directamente llegue al nodo *sink*. Debido a la limitante de energía en los sensores y a que normalmente un gran número de nodos se encuentra fuera del alcance de transmisión del *sink*, esta forma de transmisión sólo es posible con redes WSN muy pequeñas.
- **Envío directo o multisalto vía *cluster heads*.** Este mecanismo utiliza una estructura basada en conjuntos dominantes o en árboles donde hay nodos tipo *cluster head*, los cuales tienen asociados nodos fuente. Cada nodo envía sus datos a su *cluster head*, si este *cluster* tiene dentro de su alcance de transmisión al nodo *sink* entonces envía directamente los paquetes recibidos, de lo contrario los envía al siguiente *cluster head* en dirección al nodo *sink* [30]. Utilizar nodos *cluster head* permite ahorrar energía y disminuir la congestión en la red.

Los datos adquiridos por un solo sensor, en diferentes tiempos o por sensores del mismo *cluster*, pueden estar correlacionados y contener información redundante. En lugar de transmitir todos los datos, sería mejor procesar la información en los sensores intermedios o en los *cluster heads* para reducir el número de paquetes enviados al nodo *sink* y así disminuir el gasto de energía y el ancho de banda consumido. Esta técnica se llama *agregación de datos* [92] y es un proceso que consiste en agregar o combinar datos que vienen de distintos nodos fuente con el objetivo de eliminar transmisiones redundantes [43, 41, 42]. La agregación de datos normalmente involucra la fusión de información de distintas fuentes, por ello es que también se le conoce como *fusión de datos (data fusion)*. La fusión de datos también puede ser integrada con el encaminamiento, es decir, se puede tratar de encontrar rutas en las cuales se agregue o fusione un mayor número de datos (*data fusion centric routing*) [91].

En general, esta forma de transmitir los datos adquiridos implica un tráfico del tipo "muchos a uno" (múltiples nodos fuente-un nodo *sink*). Si este tipo de tráfico no es tratado con precaución, causará un alto consumo energético en toda la red y es altamente probable que a los nodos cercanos al nodo *sink* se les agote la batería rápidamente, ocasionando que la red quede desconectada. Para disminuir un poco este problema, algunas soluciones introducen nodos móviles que proactivamente se mueven alrededor del escena-

rio para recolectar los datos adquiridos por otros sensores. Utilizar nodos móviles en las redes WSN mejora la cobertura y dichos nodos pueden sustituir rápidamente a aquellos nodos cuya batería está próxima a agotarse, evitando así la desconexión. En este tipo de redes, existen tres tipos de nodos, nodos fuente (estáticos), nodos *sink* y nodos móviles, los cuales realizarán alguna tarea en específico como actuar de recolector intermedio de datos o ser un *sink* móvil o sustituir a algún nodo fuente que presente determinada falla [93].

Existen soluciones en donde sólo se utilizan nodos móviles, por ejemplo en *Pothole Patrol* [24] y *Nericell* [25] los autores utilizan vehículos equipados con varios sensores para monitorear las condiciones de las carreteras. En [94] los autores utilizan los datos del acelerómetro para clasificar los movimientos físicos del usuario que carga el teléfono. En estas soluciones los datos adquiridos son enviados a un servidor central utilizando conexiones WiFi oportunistas o la red celular si está disponible, más soluciones de este tipo serán mencionadas en el capítulo 4.

El nodo *sink* puede reasignar las tareas de monitoreo, las cuales deben ser propagadas o diseminadas a todos los sensores. En una red WSN muy pequeña, quizá el alcance de transmisión del nodo *sink* cubra toda la red y entonces sólo se requiere una transmisión para la comunicación de la tarea. En redes WSN más grandes lo anterior no es posible por lo que, los nodos intermedios deben reenviar el mensaje del nodo *sink* para que llegue a todos los sensores de la red. La técnica más simple para realizar esto es la de inundación (*flooding*), en donde cada sensor que recibe un paquete lo retransmite, utilizando esta técnica algunos sensores pueden recibir y retransmitir el paquete más de una vez. Este método es popular y es el más costoso en términos de energía. Una mejor técnica es el *broadcast*, en donde todos los nodos reenvían el paquete pero sólo lo retransmitirán una vez. También es posible sólo enviar la tarea a un grupo de nodos localizados dentro de alguna región (*geocasting*), o a un grupo de nodos identificados quizá con ciertos ID's (*multicast*). En caso de que los sensores estén organizados en una estructura de árbol, dicha estructura puede ser utilizada por el *sink* para propagar la tarea, de esta manera sólo los nodos que forman parte del árbol reenvían el paquete (*broadcasting tree*). Algunas soluciones utilizan estas técnicas de propagación para organizar la red y así recolectar datos. Esta forma de recolección de datos puede ser clasificada a su vez en tres tipos [95]:

- **Recolección de datos en dos fases (*Two-Phase Pull Diffusion*)**. En la primera fase el envío de solicitudes (desde el nodo *sink*) y de respuestas (desde los nodos fuente) son transmitidas utilizando la técnica de inundación, con lo cual se descubren múltiples rutas desde los nodos fuente hacia el *sink*. En la segunda fase, el nodo *sink* hace una elección de las mejores rutas, normalmente las de menor latencia y posteriormente los datos son enviados hacia el *sink* a través de dichas rutas [54]. Este tipo de mecanismo es adecuado cuando las redes WSN tienen muchos nodos fuente y pocos nodos *sink*.
- **Recolección de datos en una fase (*One-Phase Pull Diffusion*)**. Utilizar la técnica de inundación para el envío de solicitudes y respuestas resulta muy costoso en términos de energía cuando se tiene una gran cantidad de nodos fuente y de nodos *sink*. Por ello es que este mecanismo omite la inundación de las respuestas, las cuales son enviadas "hacia atrás", es decir, cada nodo envía su respuesta a través del nodo por el cual llegó la primera solicitud.
- ***Push diffusion***. En este tipo de mecanismo un nodo fuente envía, utilizando la técnica de inundación, la información adquirida cada vez que detecta un evento y el nodo *sink* se suscribe a los eventos de interés. Esta técnica es buena para aplicaciones en las cuales existen muchos nodos *sink*, pocos nodos fuente y estos nodos fuente generan datos ocasionalmente [55].

En las redes WSN muy grandes, donde la transmisión de datos se realiza utilizando una técnica multisalto o donde sólo hay nodos móviles, obtener una vista completa de lo que ocurre con el fenómeno monitoreado puede tomar cierto tiempo, por ello también es importante considerar qué tan rápido los datos adquiridos acerca de cierto fenómeno dejan de ser válidos para convertirse en obsoletos.

La información adquirida por los sensores puede ser almacenada quizá para construir un registro histórico o para ser analizada o consultada tiempo después de que es recolectada. Con base en dónde se almacena dicha información, los mecanismos de recolección pueden ser [95]:

- **De almacenamiento externo (ES¹⁴)**. En este tipo de métodos toda la información adquirida por los sensores es enviada hacia el *sink*, donde es almacenada para su procesamiento o posterior consulta.

¹⁴External Storage

- **De almacenamiento local (LS¹⁵)**. En esta categoría la información que adquiere cada nodo sensor se almacena localmente, por lo que no hay un gasto inmediato de energía por la transmisión de los datos. Sin embargo, cada sensor debe procesar todas las solicitudes de información enviadas por el *sink* y sólo en el caso de satisfacer dicha solicitud el nodo envía sus datos hacia el nodo *sink*.
- **De almacenamiento centrado en los datos (DCS¹⁶)**. Este tipo consiste en dos pasos, a) una vez que el sensor detecta un evento, lo etiqueta utilizando algún procedimiento previamente acordado y b) envía los datos adquiridos hacia otro nodo sensor que se encarga de almacenar los datos relacionados con dicha etiqueta. Este tipo de almacenamiento permite realizar la tarea en forma colaborativa así como una distribución en la carga de información guardada [96].

En los capítulos 3 y 4 se analizarán las alternativas de solución encontradas en la literatura que tienen una relación más directa con las estrategias para la recolección de datos que se plantean en este trabajo de investigación.

¹⁵Local Storage

¹⁶Data-Centric Storage

Capítulo 3

Método *Disagree*

3.1 Introducción

En este capítulo se presenta *Disagree*, el cual es un método de recolección de datos que logra un ahorro de energía al nivel del sensor explotando la correlación espacial de los fenómenos monitoreados. *Disagree* está orientado a redes WSN, pequeñas o medianas, integradas con sensores estáticos. Con este método es posible obtener una vista completa de lo que sucede en la red sin requerir que todos los sensores transmitan sus datos adquiridos.

La adquisición de datos con ahorro de energía ha sido un tema importante desde el inicio de las redes inalámbricas de sensores y diversos investigadores han estudiado este problema desde distintos ámbitos [3, 40]. En general, estos métodos de adquisición pueden ser clasificados considerando la organización de la red. En el capítulo 2 se mencionó que una de estas clasificaciones divide a dichos métodos en planos, jerárquicos y con conjuntos dominantes [38].

En las redes planas [3, 39, 40], cada sensor transmite individualmente sus lecturas hacia el nodo *sink* a través de una ruta multisalto. Aunque la diferencia entre las lecturas de sensores vecinos sea pequeña o nula, cada lectura es enviada; esto resulta ineficiente energéticamente debido a la gran cantidad de paquetes transmitidos.

Las redes jerárquicas [30, 3] se organizan en grupos llamados *clusters*, y en cada *cluster* hay un nodo llamado *cluster head* o cabeza de grupo; cada sensor se asocia a un *cluster head* en específico. El *cluster head* recibe e incorpora, utilizando algún método de agregación [41, 42, 43], las lecturas individuales de sus sensores asociados en un solo paquete que será enviado hacia el nodo *sink*. Esto reduce la cantidad de paquetes transmitidos, sin embargo este tipo de soluciones aún requieren que cada sensor responda a cada solicitud que le es enviada, utilizando así valiosos recursos energéticos.

Las redes basadas en conjuntos dominantes [38], organizan los sensores utilizando un conjunto dominante conectado (CDS) para formar la estructura central (*backbone*). Los nodos que no son parte del conjunto dominante son adyacentes al menos a un nodo que pertenece a dicho conjunto. En algunas soluciones basadas en este esquema, los miembros del conjunto dominante son *cluster heads*, los cuales también agregan las lecturas que les envían los demás sensores.

En algunas ocasiones se requiere que todos los nodos transmitan sus lecturas para obtener una vista o conocimiento completo de lo que sucede en la red, usualmente los métodos planos, jerárquicos y de conjuntos dominantes generan tantas respuestas como sensores existan en la red, aun cuando las variaciones entre las lecturas de sensores cercanos sea pequeña. En este trabajo se considera que debe haber una forma más eficiente para obtener una vista completa de las mediciones en la red de sensores, involucrando un menor número de transmisiones a nivel del sensor; ésta es la idea clave de nuestro método, al que hemos denominado método *Disagree*.

El método *Disagree* [33], de manera similar a algunas otras propuestas [3, 32, 97, 30, 31], intenta reducir la cantidad de datos enviados al nodo *sink*. Sin embargo, lo que hace diferente a *Disagree* es que alcanza esta meta limitando las transmisiones al nivel del sensor. Esto significa que en este método no se requiere que todos los sensores respondan a una solicitud enviada por el *sink*, aun si se solicita una vista completa de la red, ya que el *sink* puede estimar las lecturas de aquellos nodos sensores que permanecieron en silencio con algún margen de error predefinido. Lo anterior permite que los sensores ahorren energía, incrementando así el tiempo de vida de la red.

Disagree es una solución jerárquica que organiza la red utilizando una estructura de árbol, donde hay dos tipos de nodos: los nodos *árbol* son nodos que pertenecen al árbol de distribución, y los nodos *sensor* son aquellos que no pertenecen al árbol y cada uno de ellos debe asociarse a lo más con un nodo árbol. Los nodos sensores reciben *afirmaciones*¹ de datos provenientes de sensores cercanos y transmiten sus lecturas sólo si ellos están en desacuerdo con dicha afirmación; ésta es la idea clave para que se lleve a cabo una eficiente recolección de datos en *Disagree*. Debido a que en muchas aplicaciones de monitoreo hay una correlación espacial entre las lecturas de sensores vecinos, una situación común en *Disagree* es que sólo algunos sensores tendrán lecturas muy diferentes a las reportadas por el nodo árbol (en *Disagree*, el nodo árbol es similar al *cluster head*), esto permite que muchos sensores permanezcan en silencio, ahorrando importantes recursos tales como la energía y el ancho de banda.

Más que solicitar respuestas de los nodos que tengan lecturas dentro de un intervalo fijo, como lo hacen los métodos tradicionales en redes inalámbricas de sensores, *Disagree* utiliza un intervalo dinámico centrado en el valor reportado por el nodo árbol correspondiente, con una tolerancia superior e inferior definida con el parámetro llamado delta (Δ). Los sensores que tengan lecturas fuera de este intervalo se encontrarán en *desacuerdo* con la afirmación del nodo árbol y enviarán su lectura para “expresar” dicho desacuerdo. En *Disagree*, el valor de delta puede ser usado para maximizar o minimizar la resolución de las lecturas monitoreadas al nivel del sensor. Un valor grande de Δ proporciona una baja resolución espacial debido a que más nodos permanecen en silencio. De otra forma, valores pequeños de Δ permiten una alta resolución espacial, debido a que más nodos estarán en desacuerdo y transmitirán sus lecturas al nodo árbol con el que se encuentran asociados. Controlando el valor de la Δ , *Disagree* puede ser configurado como un protocolo o método de adquisición de datos en distintas resoluciones (resolución múltiple). Por ejemplo, *Disagree* podría inicialmente solicitar una resolución baja de la red usando una Δ con valor grande, después, una segunda petición puede ser enviada a regiones específicas de la red utilizando una Δ pequeña para obtener información con mayor detalle.

¹Una afirmación es un tipo de paquete de datos que los sensores envían a otros sensores cercanos, con el objetivo de “afirmar” o “confirmar” los valores monitoreados. Este tipo de paquetes será explicado con mayor detalle en la sección 3.3.2

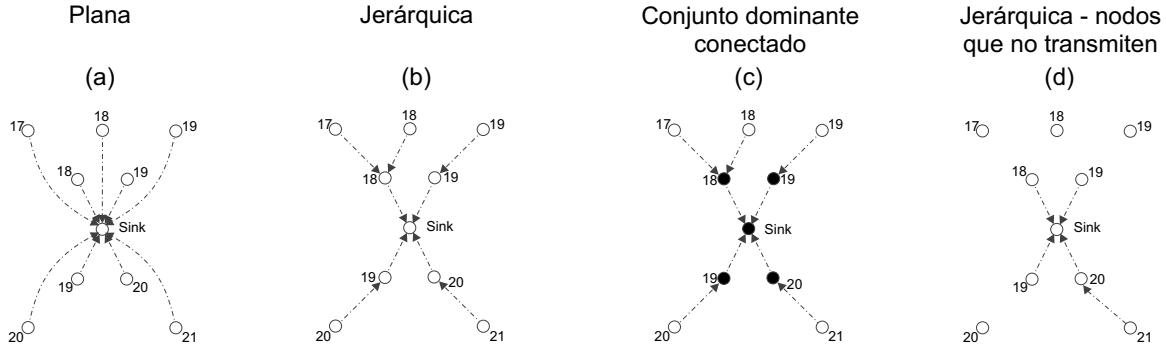


Figura 3.1: Soluciones para la adquisición de datos en las redes inalámbricas de sensores

La figura 3.1 ilustra la operación básica de cuatro métodos generales de adquisición de datos después de que el nodo *sink* solicitó una vista completa de la red. Los números en dicha figura indican ejemplos de lecturas en cada nodo. La figura 3.1(a) muestra una topología plana, donde todos los nodos envían individualmente sus lecturas hacia el nodo *sink*. La figura 3.1(b) muestra una solución jerárquica donde los nodos envían sus lecturas a los *cluster heads*, los cuales a su vez envían los datos agregados hacia el nodo *sink*. En la figura 3.1(c), los nodos que forman un conjunto dominante conectado (CDS) están indicados en color negro y los nodos sensores en color blanco. Aquí los sensores envían sus lecturas a su vecino más cercano en el conjunto dominante, el cual a su vez agrega y reenvía los datos hacia el nodo *sink*, siendo este caso muy similar a la solución jerárquica. Finalmente, la figura 3.1(d) muestra una solución jerárquica donde hay nodos que permanecen en silencio, es decir, no transmiten sus lecturas hacia el nodo *sink*, pero es posible estimar las lecturas de esos nodos que no transmiten con un margen de error; como se puede observar, *Disagree* pertenece a esta última categoría. El desempeño de *Disagree* mejora si existe una alta correlación espacial en el fenómeno a monitorear. Esto se logra cuando las diferencias de las lecturas de los nodos sensores, con respecto al valor central afirmado por el nodo árbol, se vuelven más pequeñas; lo cual es una propiedad encontrada en muchos fenómenos físicos.

De manera similar, el desempeño de *Disagree* se incrementa si se colocan más nodos en la red. Lo anterior ocurre porque al haber más sensores en una misma área, se incrementa la probabilidad de que más sensores vecinos monitoreen lecturas similares y menos nodos estarán en desacuerdo. Esto se revisará a detalle en la sección 3.5. Es importante

mencionar que varias aplicaciones en este tipo de redes inalámbricas requieren de un denso despliegue de sensores para lograr una mejor cobertura y por ende una mayor resolución espacial. Sin embargo, aun cuando la densidad de nodos en la red sea baja, *Disagree* se desempeña mejor en comparación con las técnicas jerárquicas tradicionales, esto será demostrado en la sección de implementación y evaluación, véase la sección 3.6.

3.2 Trabajo relacionado

A continuación se discute cómo *Disagree* contrasta con el trabajo relacionado en el tema. *Directed Diffusion* [54], es uno de los métodos pioneros en la adquisición de datos, en el cual las solicitudes son enviadas a la red utilizando la técnica tradicional de inundación [65], y los sensores responden con sus lecturas utilizando la ruta por la cual llegaron dichas solicitudes. Más que optimizar el consumo de energía en los sensores, el énfasis principal de *Directed Diffusion* es enviar los datos recolectados a través de las rutas encontradas hacia el *sink*, pero en dirección contraria en la que llegó la solicitud. La técnica tradicional de inundación se utiliza comúnmente para propagar las solicitudes de datos y requiere que cada nodo reenvíe el paquete recibido al menos una vez, esto genera muchas transmisiones redundantes, consumiendo así recursos valiosos [98]. *Gossiping* [64] es un derivado de la técnica de inundación [65], en el cual una vez que algunos nodos reciben las solicitudes, seleccionan aleatoriamente a algún nodo vecino y sólo al nodo seleccionado le reenvían dicha solicitud. Esta técnica evita los problemas de la inundación pero, debido a que cada nodo sólo reenvía el paquete a un nodo vecino, provoca que tome un mayor tiempo para propagar la solicitud a todos los nodos en comparación con *Directed Diffusion*. En *Rumor routing* [56], los sensores construyen rutas hacia los eventos, de tal forma que el envío de una solicitud puede seguir dichas rutas en lugar de inundar toda la red con paquetes de solicitud. Esta operación ahorra energía durante el proceso de la solicitud, sin embargo, se requiere de energía adicional para crear las rutas. De esta forma, al final el ahorro energético depende de las características del fenómeno que se requiere monitorear.

Los métodos jerárquicos utilizan un nodo llamado *cluster head*, el cual recolecta las lecturas individuales de los nodos asociados y agrega la información recolectada en un solo paquete que será enviado hacia el nodo *sink*. En general, cada sensor se asocia a cier-

to *cluster head* y los *cluster heads* pueden formar múltiples niveles de jerarquía. En este tipo de soluciones hay un ahorro importante de energía pues en lugar de enviar muchas respuestas individuales hacia el nodo *sink*, como en las redes planas, se envía una sola respuesta por región. Entre las soluciones de este tipo se encuentran las siguientes: En LEACH² [30], los *cluster heads* recolectan datos localmente utilizando TDMA³ y rotan el papel de *cluster head* entre los sensores que forman parte del mismo *cluster* para evitar que se agote rápidamente la energía de dichos nodos. En [99], los autores proponen un algoritmo llamado DDCHS⁴, el cual selecciona dinámicamente como nuevo *cluster head* al nodo que se encuentra más cercano al *cluster head* seleccionado previamente y que tenga el mayor número de vecinos. En [100], los autores proponen HEED⁵, donde la selección del *cluster head* se basa principalmente en la energía residual de cada nodo para balancear el consumo de energía (se asume que todos los nodos son equipados inicialmente con la misma cantidad de energía). En H-HEED⁶ [101], los autores consideran la heterogeneidad en términos de la energía en el nodo y rotan el rol del *cluster head*. En [102], los autores proponen un protocolo de recolección de datos eficiente, en cuanto al ahorro de energía, basado en una estructura de árbol EEDCP-TB⁷. Este método escoge a los nodos que tengan una mayor energía residual para balancear el consumo de energía. En [103], los autores proponen un algoritmo para balancear cargas y así reducir el consumo de energía en redes inalámbricas de sensores heterogéneos. En [104] los autores proponen DGLB, el cual clasifica a los nodos sensores en diferentes capas basadas en cuántos saltos hay entre el nodo *sink* y los nodos sensores. Este algoritmo permite que los sensores presenten un consumo de energía uniforme. Una característica común de todos estos métodos es que se rota el papel del *cluster head*, pero no explotan la correlación de los datos, ya sea espacial o temporal, a nivel de sensor porque todos los nodos envían sus lecturas al *cluster head* local, el cual a su vez reenvía tales lecturas hacia al nodo *sink*, ya sea en varios paquetes o agregadas en un solo paquete.

Entre los métodos jerárquicos que explotan la correlación temporal se encuentran TEEN⁸ [59], el cual es similar a LEACH, excepto que el *cluster head* proporciona dos umbrales

²Low-Energy Adaptive Clustering Hierarchy

³Time Division Multiple Access

⁴Density and Distance based Cluster Head Selection

⁵Hybrid Energy Efficient Distributed

⁶Heterogeneous-Hybrid Energy Efficient Distributed

⁷Energy Efficient Data Collection Protocol - Tree Based

⁸Threshold sensitive Energy Efficient

(duro y suave). En caso de que la nueva lectura, tomada por el sensor sea más grande que el umbral duro, y la diferencia entre dicha lectura y la anterior sea más grande o igual que el umbral suave, entonces el sensor transmite esta nueva lectura a su *cluster head*. El método TEEN está orientado a aplicaciones de monitoreo reactivo, su desempeño no es bueno en aplicaciones proactivas donde se requiere que las lecturas sean enviadas periódicamente. Por lo anterior, los autores desarrollaron APTEEN⁹ [105], el cual combina TEEN con la funcionalidad de TDMA para proporcionar un buen desempeño tanto en aplicaciones reactivas como proactivas. En [61] los autores proponen un protocolo llamado SHRP¹⁰ cuyo principal objetivo es encontrar una ruta que incremente el tiempo de vida de la red, explotando la correlación temporal en los sensores. En este trabajo, cada sensor verifica qué tanto ha cambiado la lectura anterior respecto a la actual para decidir si se envía o no el nuevo valor al *cluster head*. Es importante mencionar que no se comparó *Disagree* con ninguno de estos métodos porque *Disagree* explota principalmente la correlación espacial.

Los métodos jerárquicos que explotan la correlación espacial y temporal en las redes inalámbricas de sensores generalmente utilizan una estructura jerárquica en la cual los *cluster heads* aplican técnicas de compresión sobre los datos recolectados para así reducir el tamaño de la respuesta. En estos métodos se puede considerar un esquema de resolución múltiple, en el cual el *sink* pueda establecer la resolución espacial deseada. En general, una resolución alta implica un mayor consumo de energía debido a que más datos requieren ser enviados hacia el nodo *sink*. En MRCQ¹¹ [106], los autores utilizan una estructura jerárquica basada en capas, los datos reportados de las capas inferiores son comprimidos por el nodo que se encuentra en una capa superior utilizando técnicas tradicionales de compresión de datos [107, 108]. En este método, sólo la información con baja resolución es reportada hacia el nodo *sink* para lograr un ahorro de energía. Sin embargo, la información con alta resolución es almacenada en los nodos de la capa superior para enviarla en caso de que llegue un requerimiento con una resolución mayor. DIMENSIONS [109] utiliza la técnica de *waveletes* [110] para comprimir datos que tiene largos periodos de vida (*long-term data traces*). Este método puede ser usado para obtener información de la red con alta o baja resolución. Se hace énfasis en que *Disagree*,

⁹Adaptive Periodic Threshold-sensitive Energy Efficient

¹⁰Simple Hierarchical Routing Protocol

¹¹Multi Resolution Compression and Query framework

a diferencia de estos métodos, no utiliza técnicas de compresión para reducir el número de paquetes transmitidos, aunque puede incorporar dichas técnicas para disminuir aún más la cantidad de información que será enviada al nodo *sink*.

Los protocolos orientados a la detección de eventos en la red usualmente explotan la correlación temporal. En [111] los autores proponen un método llamado *Ken* para ahorrar energía a nivel del sensor. En este trabajo, los sensores utilizan un modelo de probabilidad para predecir la siguiente lectura basándose en lecturas previas. Cuando una nueva lectura es diferente de la lectura esperada, los sensores proactivamente envían la nueva lectura hacia el nodo *sink*. En [112, 32], sólo los miembros de un conjunto representativo de nodos comunican sus lecturas hacia el *sink*. Sin embargo, estos métodos son sensibles a las variaciones de los datos en el tiempo, por lo que requieren que todos los nodos envíen periódicamente sus lecturas al *cluster head* o al nodo *sink* para actualizar los parámetros de la próxima solicitud. Adicionalmente, estos métodos proporcionan una sola resolución espacial de los valores monitoreados en la red. Opuesto a estos protocolos, *Disagree* no es sensible a las variaciones en la correlación de datos en el tiempo, de tal forma que la solicitud, que viaja a través del árbol, no requiere ser actualizada. Por otro lado, *Disagree* proporciona vistas de la red con diferentes niveles de resolución dependiendo de la aplicación particular para la que se esté utilizando.

En la categoría de los métodos jerárquicos que incluyen nodos que no transmiten, y a la cual pertenece *Disagree*, se encuentran las siguientes propuestas: SMRDD¹² [113] divide un campo de sensores en pequeñas regiones llamadas *grids* o retículas, donde el tamaño de la retícula corresponde al nivel deseado de resolución espacial de datos. En este método hay dos tipos de nodos: *pixel points*, que son nodos similares a los *cluster heads* y nodos *home*, los cuales están asociados a los *pixel points*. Cada retícula tiene un *pixel point* y uno o más nodos *home*. Las peticiones de información son enviadas a los *pixel points* a través de las rutas del árbol, compuesto de un nodo *sink* y de todos los *pixel points*. SMRDD no puede obtener o estimar las lecturas de los sensores que no pertenecen a un tamaño de retícula en particular, es por eso que recolecta menos información en comparación con *Disagree*. Para comparar SMRDD con *Disagree*, SMRDD necesitaría obtener o estimar las lecturas de todos los sensores y esto ocasionaría que todos los sensores transmitieran su información; resultando un comportamiento similar a

¹²Spatial-based Multi-Resolution Data Dissemination

los métodos basados en *cluster heads*. En CAG¹³ [31], los autores obtienen datos de una red inalámbrica de sensores utilizando un subconjunto de nodos representativos en la red. CAG opera en dos fases llamadas petición o requerimiento de información (*query*) y respuesta. Durante la primera fase, una petición de información es diseminada a través de la red utilizando la estructura de árbol, integrada por *cluster heads* y nodos tipo puente (*bridge*). Basándose en la lectura del *cluster head* (CR), la lectura del nodo (MR) y un umbral (τ), un nodo decide unirse a un *cluster* sólo si se cumple que $MR < CR \pm CR \times \tau$, de lo contrario el nodo se declara a sí mismo como un nuevo *cluster head*, diseminando una nueva petición con su propia lectura a los nodos vecinos. Durante la fase de respuesta, sólo los *cluster heads* transmiten sus lecturas hacia el nodo *sink*. En [114, 115], cada nodo escucha las lecturas de todos sus vecinos para determinar la redundancia de su propia lectura. Este método explota la correlación espacial para reducir el número de paquetes de datos transmitidos, pero frecuentemente, todos los nodos deben transmitir mensajes de control para determinar dicha redundancia.

A continuación se mencionan algunas soluciones que utilizan un conjunto dominante conectado (CDS) para construir el *backbone* de la red. En [97] los autores proponen un algoritmo para construir un CDS basándose en la correlación espacial de los datos. Este protocolo tiene una fase inicial, en la cual todos los nodos transmiten sus lecturas a sus vecinos para crear el CDS. Después de esto, cada nodo evalúa si su lectura actual puede ser inferida como una combinación lineal de las lecturas de los nodos vecinos; si se satisface esta condición, el nodo informa a los otros nodos que no responderá a futuras peticiones de información. Durante la segunda fase, sólo los nodos que forman parte del CDS envían sus lecturas hacia el *sink* y con esto se reduce el número de paquetes transmitidos. Este protocolo tiene una etapa de mantenimiento para construir un nuevo CDS que se adapte a los cambios que presenten las nuevas lecturas. Este método no puede ser utilizado para obtener múltiples resoluciones porque sólo los nodos en el CDS transmiten sus lecturas y con esto sólo se puede obtener una resolución; debido a esto y a que se requiere un mantenimiento periódico donde todos los nodos transmiten, este método no es directamente comparable con *Disagree*. Una situación similar ocurre con el método presentado en [116], donde los autores proponen la selección de unos cuantos nodos para formar un *backbone* virtual que soporte la creación de rutas y monitoreo.

¹³Clustered AGregation

A manera de resumen, la mayoría de los métodos jerárquicos que proporcionan resolución múltiple propuestos para las WSN explotan la redundancia de los datos a nivel de *cluster head*, por lo que requieren que los sensores transmitan sus lecturas individuales al *cluster head* con el que se encuentran asociados. La diferencia clave de estas técnicas con respecto a *Disagree*, es que *Disagree* explota la redundancia espacial de los datos a nivel del sensor, estimando las lecturas de aquellos nodos que no transmiten, lo que permite que los sensores eviten realizar transmisiones de datos innecesarias y así poder ahorrar energía.

Para comparar el desempeño de *Disagree*, se seleccionó LEACH porque es un ejemplo de un método jerárquico que incluye un algoritmo para rotar el papel de *cluster head* y, en este punto en particular, es similar a *Disagree*. También se seleccionó CAG porque este protocolo, de manera similar a *Disagree*, explota la redundancia espacial de los datos a nivel del sensor, permitiendo que algunos nodos permanezcan en silencio y también estima las lecturas de los nodos que no transmiten. Sin embargo, es importante enfatizar que CAG fuerza periódicamente a todos los sensores a transmitir para actualizar los parámetros de su solicitud (*query tree*). En este aspecto difiere de nuestra propuesta. A pesar de esta diferencia, se consideró que la operación de CAG es muy cercana a la de *Disagree*. No se comparó *Disagree* con más protocolos porque no se encontraron propuestas que compartan características con *Disagree*. Los resultados de comparar *Disagree* con LEACH y CAG se presentan en la sección 3.6.

3.3 Descripción de *Disagree*

Disagree [33] es un método para recolectar datos en las redes inalámbricas de sensores, su principal objetivo es el ahorro de energía en los sensores explotando la correlación espacial que poseen los fenómenos monitoreados por dichos sensores.

Considerando el tipo de aplicación, las redes inalámbricas de sensores pueden clasificarse en proactivas y reactivas [59]. En la redes proactivas los sensores monitorean periódicamente el fenómeno de interés y reportan los datos hacia un nodo central. En las reactivas los sensores también realizan un monitoreo periódico, pero sólo reportan los datos obte-

nidos cuando la diferencia entre las lecturas actuales y las anteriores rebasen el umbral determinado por la aplicación. El método *Disagree* está optimizado para las aplicaciones de monitoreo proactivo y su operación consiste en cinco etapas denominadas *estructura*, *expansión*, *disagree*, *respuesta* y *rotación*; cada una de estas etapas son detalladas a continuación.

3.3.1 Estructura

El objetivo de esta etapa es organizar la red utilizando una estructura jerárquica de árbol [117], esto se lleva a cabo al iniciar la operación de *Disagree* y sólo ocurre una vez. Para obtener dicha estructura se usa el algoritmo de *Prim* [117], el cual utiliza la máxima distancia entre los nodos como el peso mínimo. En la estructura creada hay dos tipos de nodos: *nodos árboles (TN)* y *nodos sensores (SN)*. Los *TN* son nodos que pertenecen al árbol de distribución, con el caso especial del nodo TN_0 , el cual es la raíz del árbol y también se le conoce como nodo *sink*. Los *SN* son nodos que no pertenecen al árbol.

Durante la construcción del árbol, cada nodo selecciona a otro nodo para asociarse con él. Una vez que el árbol está construido, si un nodo tiene nodos asociados, éste será un nodo árbol (*TN*), de lo contrario será un nodo sensor (*SN*). En la figura 3.2(a) se muestra un conjunto de sensores estáticos que no están organizados, en la figura 3.2(b) se ilustra la estructura de árbol que se obtiene a partir de la aplicación del algoritmo de *Prim*. Por simplicidad, se asume que cada nodo conoce su ubicación (utilizando GPS o cualquier otro método de localización [118]).

3.3.2 Expansión

Una vez que el árbol mínimo está construido, ocurre la etapa de expansión. En esta etapa el nodo *sink* difunde (utilizando la técnica de *broadcast*) un paquete a través del árbol hasta cubrir toda la red. Este paquete se denomina *paquete de afirmación* y tiene la siguiente estructura: $Assertion(VARIABLE, READING, \Delta)$, donde *VARIABLE* es el fenómeno que está siendo monitoreado (temperatura, humedad, luz, etc.), *READING* es la lectura o valor observado por el *TN* que está enviando el paquete y Δ es utilizado para establecer un intervalo. Al recibir el paquete, cada *TN* lo actualiza con su propia lectura antes de reenviarlo; los *SN* que reciben dicho paquete de afirmación no lo retransmiten.

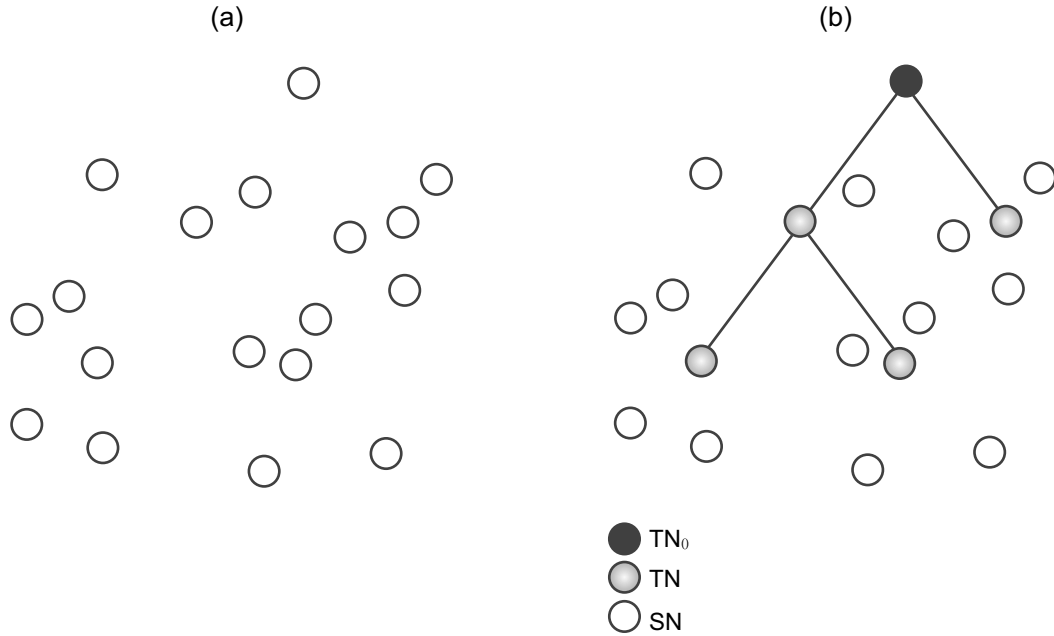


Figura 3.2: Etapa de estructura

Por ejemplo, en la figura 3.3 el nodo TN_0 difunde el paquete $Assertion(temp, 28, 3)$ o $A(temp, 28, 3)$, éste llega a sus nodos vecinos, TN y SN , sólo los TN lo retransmitirán, pero antes de hacerlo cada TN lo actualiza con su propia lectura. En la misma figura 3.3, se observa que TN_A recibe el paquete $A(temp, 28, 3)$ (enviado por el nodo *sink*), lo actualiza con su propia lectura y retransmite $A(temp, 30, 3)$.

3.3.3 *Disagree*

Como ya se mencionó, los TN difunden paquetes (*Assertion*) para afirmar los valores observados y establecer un intervalo. Cada nodo sensor que escucha dicha afirmación deberá expresar su “opinión”, verificando si su propio valor monitoreado se encuentra o no en el intervalo establecido por el paquete *Assertion* y es aquí donde surge la idea fundamental de *Disagree*, la cual consiste en que si el valor del nodo está dentro del intervalo entonces este nodo está de acuerdo con la afirmación y no realizará ninguna transmisión. Por el contrario, si su valor está fuera del intervalo, el nodo no está conforme con la afirmación y expresará su desacuerdo (*disagree*) transmitiendo su propio valor en un paquete denominado *paquete de desacuerdo*.

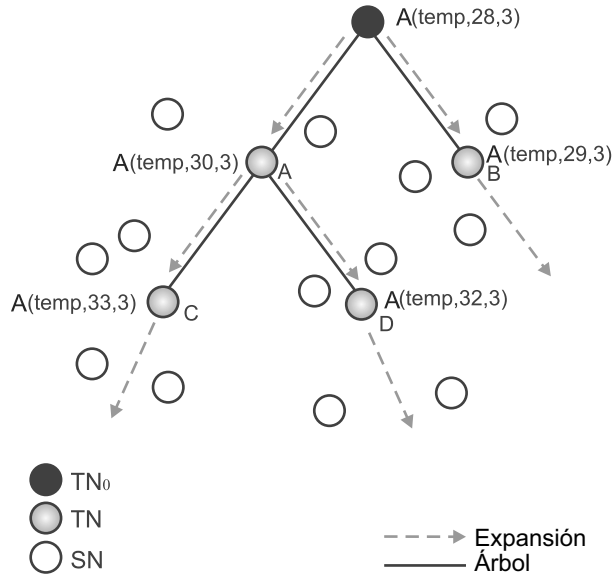


Figura 3.3: Etapa de expansión

Para detallar el funcionamiento de esta etapa se utiliza el ejemplo ilustrado en la figura 3.4, aquí sólo los sensores que tengan lecturas fuera del intervalo $[temp \pm 3]$, establecido en el paquete de afirmación previamente recibido, estarán en desacuerdo y transmitirán sus lecturas al TN con el que se encuentran asociados. En esta figura los TN y SN se indican con letras y las lecturas de cada SN son denotadas con números. Se puede observar que los sensores SN_E , SN_I y SN_Q tienen valores fuera del intervalo, por ello envían sus lecturas (paquetes de desacuerdo) a su nodo asociado TN , utilizando una transmisión *unicast* (línea punteada). Todos los otros SN (marcados con una x en la figura) permanecen en silencio. A pesar de que los sensores no transmiten su lectura a su TN asociado, es fácil para cada TN inferir que las lecturas de todos aquellos nodos que permanecieron en silencio están dentro del intervalo $[temp \pm 3]$.

Cuando los SN tienen lecturas muy cercanas a las lecturas enviadas por su TN o Δ es grande, se espera que pocos SN estén en desacuerdo con el paquete de afirmación, esto permitirá que la mayoría de los nodos sensores permanezcan en silencio ahorrando energía. Si la diferencia entre los valores observados por los SN y su TN es grande, o el valor de Δ es pequeño, un mayor número de SN estarán en desacuerdo con la afirmación de su TN y esto los forzará a enviar sus lecturas.

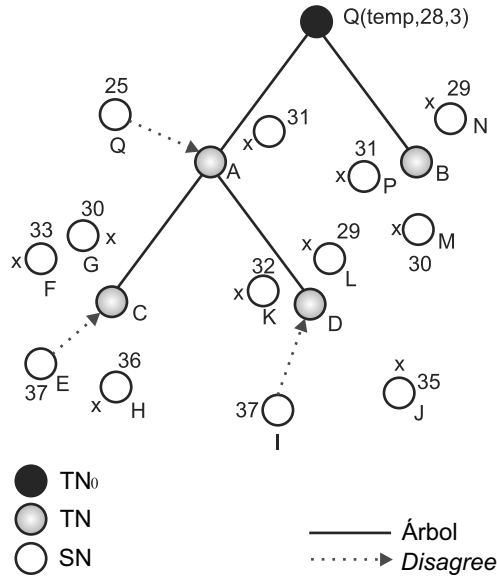


Figura 3.4: Etapa *Disagree*

Como se mencionó, la operación de *Disagree* causa que algunos sensores permanezcan en silencio, por lo que es necesario distinguir los sensores que están en un silencio “normal” de aquellos que están en silencio porque ya no tienen batería (están apagados). Para lograr esto, en *Disagree* cada sensor envía periódicamente un paquete de control al nodo TN con el que está asociado para reportarle cuál es su nivel de batería. Una vez que un TN detecta que a algún nodo se le acabó la batería, éste envía una notificación al nodo *sink*.

3.3.4 Respuesta

Durante la etapa de respuesta, la información recolectada en cada TN es transmitida hacia el nodo *sink*., ver la figura 3.5. Después de que los TN reciben los paquetes de desacuerdo enviados por algunos de sus SN asociados, cada TN toma los valores incluidos en dichos paquetes y con todos estos datos construye a su vez un sólo paquete denominado *paquete de datos*, mismo que será enviado hacia el nodo *sink*, comenzando así la etapa de respuesta. El envío de paquetes de datos es iniciado por los TN ubicados en el borde u orilla de la red de sensores. Los TN determinan que se encuentran en la orilla de la red utilizando algún método para este fin, ya sea centralizado o descentralizado [119, 120, 121].

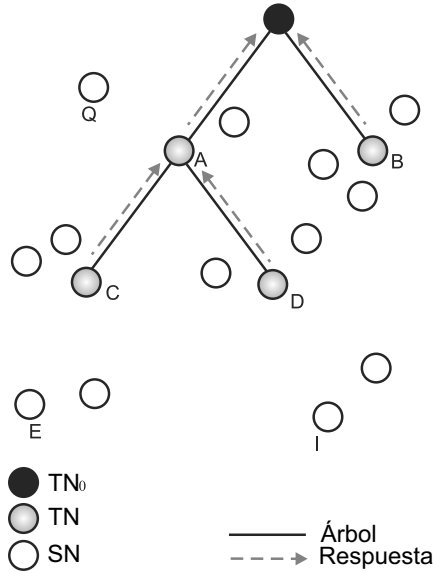


Figura 3.5: Etapa de Respuesta

Para reducir el tamaño de los datos que se incluyen en el paquete de datos, los TN pueden utilizar algún método de agregación o compresión que explote la redundancia espacial y temporal de los mismos como [42, 57, 41, 122, 43, 123], sin que esto afecte el comportamiento fundamental de *Disagree*. En nuestra implementación, los TN agregan los datos recolectados utilizando una simple concatenación de los ID y las lecturas recibidas de los SN , agregando también su propio ID y su lectura al paquete de datos que es enviado hacia el nodo *sink*.

Para mantener al mínimo el número de paquetes transmitidos en esta etapa, cuando un TN tiene dos o más TN asociados “hacia abajo”, el TN espera hasta que cada hijo TN envíe su información, ver la figura 3.5. Una vez que la información es concatenada en el nodo árbol, ésta es dirigida o enviada hacia el nodo *sink*.

Cuando el nodo *sink* ha recibido los datos de todos los TN , éste puede tener una visión completa de lo que sucede en la red, como se ilustra en la figura 3.6. En esta figura se muestra un ejemplo en el cual el nodo *sink* recibe datos de los nodos TN_A , TN_B , TN_C y TN_D . Con esta información el *sink* reconstruye el comportamiento de la red de sensores. Como se aprecia en este ejemplo, sólo los nodos SN_E , SN_I y SN_Q enviaron sus lecturas a sus TN asociados. Para todos los otros nodos en la red, el nodo *sink* sólo

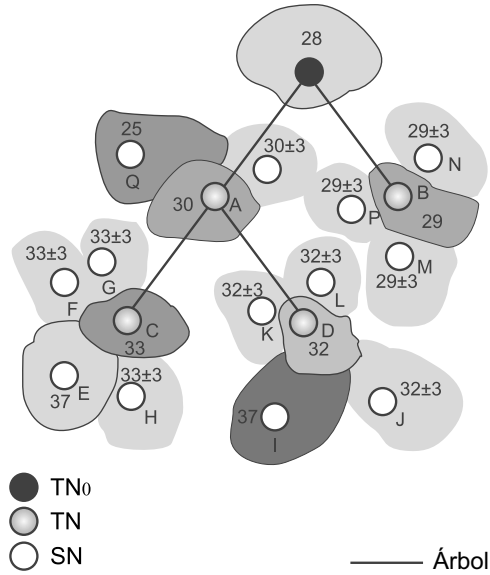


Figura 3.6: Reconstruyendo una vista completa del comportamiento del fenómeno de interés

tiene valores estimados. En el algoritmo 1 se muestra la operación completa de *Disagree*. La complejidad de este algoritmo propuesto es de $O(2n)$.

3.3.5 Rotación

Durante la operación de *Disagree* los *TN* realizan las siguientes actividades:

- Actualización y reenvío de paquetes de afirmación (*Assertion*).
- Recolección y agregación de paquetes de desacuerdo de los nodos sensores en la tercera etapa.
- Reenvío de los paquetes de datos recolectados por los otros nodos árboles hacia el sink durante la etapa de respuesta

Dichas actividades ocasionan que los *TN* transmitan con mayor frecuencia que los *SN*, esto provoca que consuman más energía. Para evitar la muerte prematura de los *TN*, se pueden utilizar distintas técnicas como equipar a los nodos árboles con baterías más grandes o rotar periódicamente el rol del nodo árbol entre los otros nodos de la red. En *Disagree* se decidió implementar un algoritmo de rotación.

Algorithm 1 Disagree(N, TN, SN, S)

/*N: all nodes*/

/*TN: nodes in tree*/

/*SN: nodes not in tree*/

/*S: sink node*/

```
1:  $S$  broadcasts an assertion packet ( $pkt_a$ )
2: for  $i = 1 \rightarrow N$  do
3:   if  $N_i \in TN$  then
4:     Receives  $pkt_a$  from its father  $TN_i$ 
5:      $fathervalue = value$ 
6:     Updates  $pkt_a$  with its own reading ( $value_{N_i}, \delta$ )
7:     Broadcasts  $pkt_a$ 
8:   else
9:     Receives  $pkt_a$  from its father  $TN_i$ 
10:  end if
11: end for
12: for  $i = 1 \rightarrow N$  do
13:   if  $N_i \in SN$  then
14:      $value =$ sensing the phenomenon
15:     if  $value > fathervalue + \delta$  or  $value < fathervalue - \delta$  then
16:       Transmits its  $value$  to its father  $TN_i$ 
17:     else
18:       Do not transmit its  $value$ 
19:     end if
20:   else  $\{N_i \in TN\}$ 
21:     if  $TN_i$ haschild( $s$ ) then
22:       Waits certain time for readings
23:       Transmits all received readings to sink
24:     else  $\{N_i$ hasasachildanode  $\in TN\}$ 
25:       Waits until receives reading from its  $TN_i$  child
26:       Transmits all received readings to sink
27:     end if
28:   end if
29: end for
```

Este algoritmo se lleva a cabo en intervalos periódicos denominados “épocas” y su operación es la siguiente: todos los nodos mantienen un contador (cTN) que indica cuántas veces el nodo ha sido nodo árbol. El nodo *sink* inicia el proceso de rotación inundando la red con un pequeño paquete llamado *rotatetree*, dicho paquete incluye dos campos:

ID es el identificador del nodo y el cTN . Al recibir el paquete *rotatetree*, cada nodo almacena la información para mantener un registro de todos los ID y cTN que escuche de sus nodos vecinos; y antes de retransmitir dicho paquete lo actualiza con su propio ID y su cTN . Cada nodo selecciona como su nuevo vecino al nodo que tenga el menor valor en cTN . En caso de que dos o más nodos tengan el mismo cTN , la selección es aleatoria. Finalmente, cada nodo envía un paquete *unicast* al nodo seleccionado para asociarse con él. La operación completa se muestra en el algoritmo 2. La figura 3.7 ilustra un ejemplo del algoritmo de rotación. La figura 3.7(a) muestra un árbol en la época uno, mientras que la figura 3.7 (b) muestra el árbol resultante de la aplicación del algoritmo de rotación.

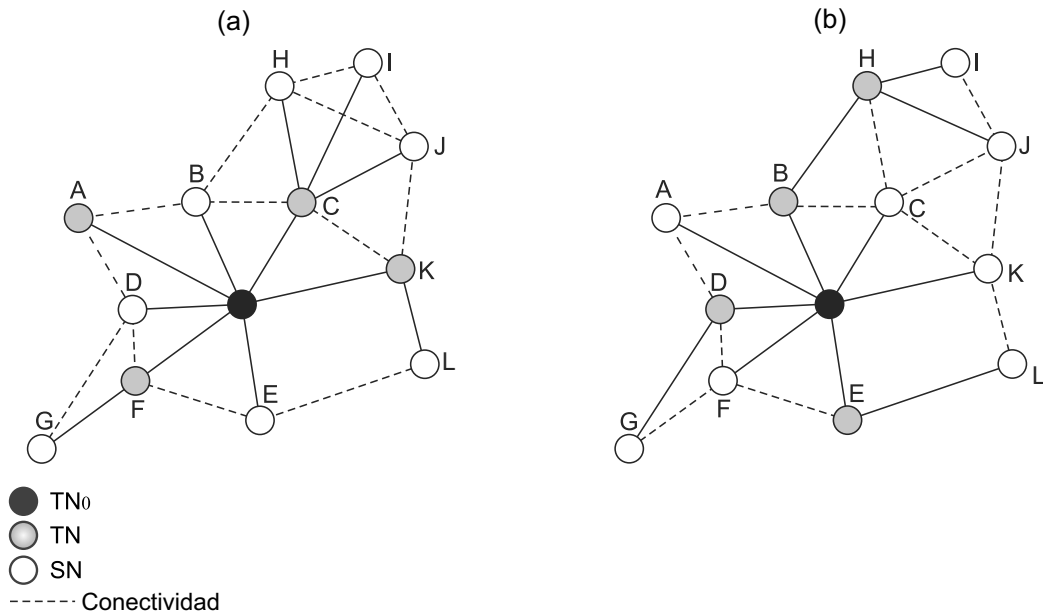


Figura 3.7: Rotación del árbol. (a) Un árbol en la época uno y (b) El árbol resultante de aplicar el algoritmo de rotación.

3.4 El valor de Δ

Un aspecto importante en *Disagree* es cómo establecer el valor de Δ . Esta decisión depende de la resolución necesaria y el error permitido para cada aplicación en particular. Con un valor pequeño para Δ , mayor será la resolución de la información y el error y ahorro de energía serán pequeños. Con un valor grande de Δ , la resolución será menor y el error y ahorro de energía serán mayores.

Algorithm 2 Tree rotation($N, TN, SN, N, epoch, leveltree$)

/*N: all nodes*/

/*TN: nodes in tree*/

/*SN: nodes are not in tree*/

/*S: sink node*/

/*epoch: number of rounds*/

/*leveltree: number of levels in tree*/

```
1: for  $i = 1 \rightarrow epoch$  do
2:   for  $j = 1 \rightarrow leveltree$  do
3:      $S$  broadcasts  $rotatetree(ID, cTN)$ 
4:     Each node receives  $rotatetree$  and saves the information
5:     Each node updates this packet with its own ( $ID, cTN$ ) and broadcasts it
6:     Each  $SN$  should associate with the  $TN$  that has a minimum  $cTN$ 
7:   end for
8: end for
```

3.5 Explotando la correlación espacial

“Todas las cosas están relacionadas, pero las cosas cercanas están más relacionadas que las cosas distantes” [124], esta observación estadística implica que la correlación de los datos se incrementa cuando la separación espacial disminuye; *Disagree* explota este fenómeno.

En esta sección se explica cómo *Disagree* explota la correlación espacial de los datos en las redes inalámbricas de sensores. También se describe un método sencillo para calcular si un sensor estará o no en desacuerdo con un paquete de afirmación; se discutirá cómo este método puede ser utilizado para ahorrar energía y se realizará una comparación de *Disagree* con el esquema tradicional basado en *cluster-head*.

Para detallar lo anterior se utiliza el escenario ilustrado en la figura 3.8 (parte superior), donde una red estática de sensores monitorea cierta variable de interés. En esta figura, la superficie representa el valor del fenómeno que está siendo monitoreado en cada punto de la red, dicho valor se expresa como $Reading(x, y) = Z_{TN}$, donde Z_{TN} es la altura. Nos enfocaremos en el nodo TN_3 el cual, después de haber recibido un paquete de afirmación (*Assertion*) y haber actualizado el campo *READING* con su propia lectu-

ra, difunde dicho paquete como $Assertion(METRIC, READING_{TN_3}, \Delta)$. Para TN_3 , estamos interesados en las dos siguientes curvas de nivel:

$$f(x, y) = READING_{TN_3} + \Delta \quad (3.1)$$

$$f(x, y) = READING_{TN_3} - \Delta \quad (3.2)$$

Las ecuaciones 3.1 y 3.2 corresponden a las curvas de nivel de $Reading(x, y)$, teniendo como alturas $READING_{TN_3} + \Delta$ y $READING_{TN_3} - \Delta$, respectivamente (ver figura 3.8 arriba). Como se mencionó anteriormente, sólo los SN asociados cuya lectura esté fuera del intervalo $READING_{TN_3} \pm \Delta$ deben responder a la afirmación, mientras que los otros SNs asociados permanecerán en silencio. Está situación se muestra en la figura 3.8 (parte inferior), donde sólo los SN asociados ubicados dentro del área gris ($A_{reply}^{TN_3}$), que a su vez se encuentren dentro del área de cobertura de TN_3 y las dos curvas de nivel, deberán responder a la petición. Los nodos ubicados en las áreas blancas ($A_{silent}^{TN_3}$) permanecen en silencio. En la figura se puede observar que $A_{silent}^{TN_3} = A_{coverage}^{TN_3} - A_{reply}^{TN_3}$, donde $A_{coverage}^{TN_3} = \pi R^2$ es el área cubierta por el alcance de transmisión de TN_3 , con un radio de transmisión de R metros. De manera similar, asumiendo una densidad uniforme de los nodos, la probabilidad de que un SN asociado a TN_3 permanezca en silencio, después de recibir un paquete de afirmación, puede ser aproximada como:

$$Probability_{silent} = 1 - Probability_{reply} \approx \frac{A_{silent}^{TN_3}}{A_{coverage}^{TN_3}} \quad (3.3)$$

Las figuras 3.9 (a,b,c) ilustran las áreas de cobertura de la red, teniendo 9 TNs (incluyendo el nodo *sink*), cada TN es responsable de administrar cierto número de SNs . Dichas figuras muestran el área gris resultante asociada a cada TN para una Δ de 1, 5 y 10 unidades de acuerdo a la superficie mostrada en la figura 3.8, respectivamente.

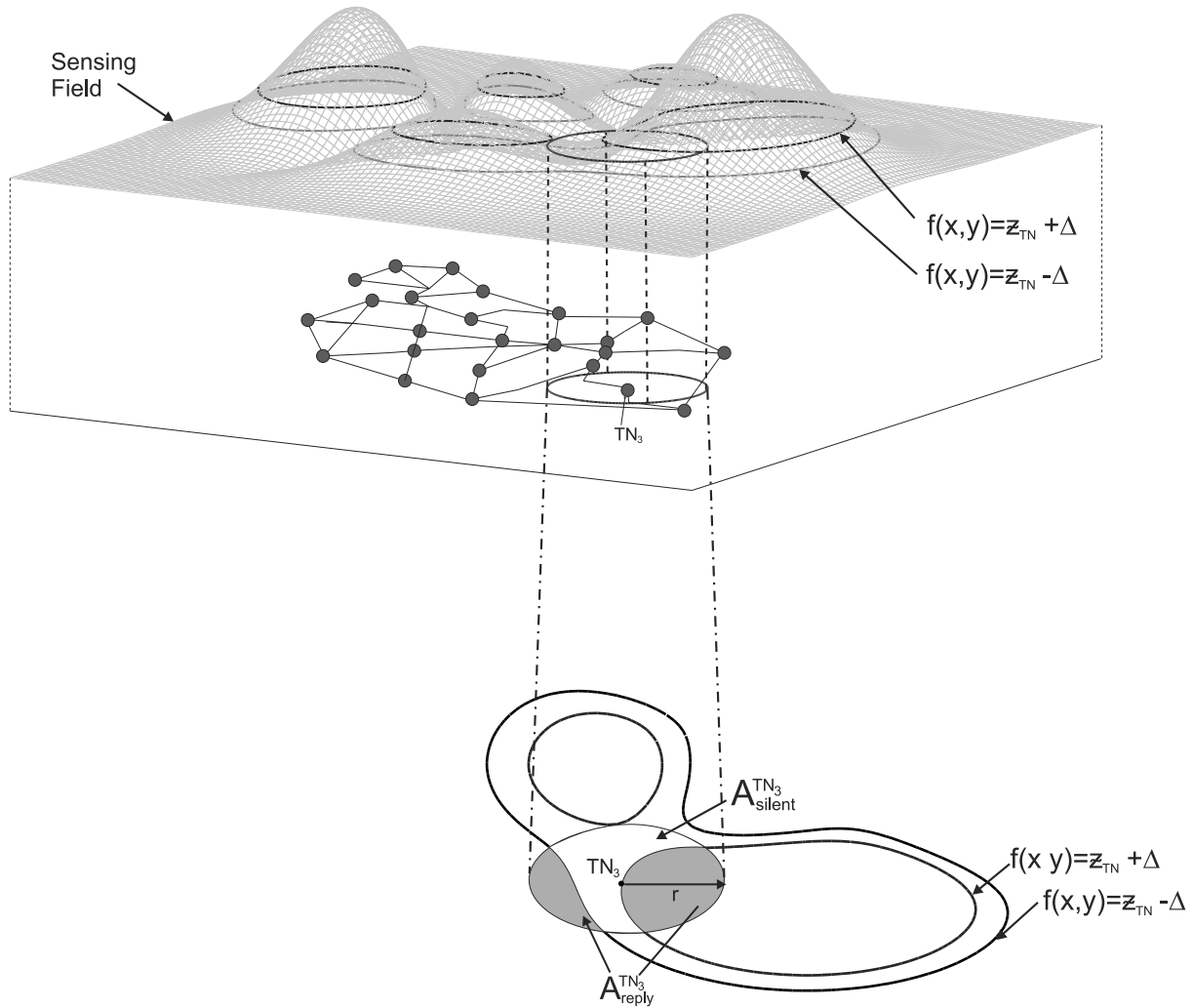


Figura 3.8: (Superior) Una red de sensores sobre un campo monitoreado. La superficie representa los valores del fenómeno en la red, las dos curvas de nivel están indicadas con $f(x, y) = Z_{TN} + \Delta$ y $f(x, y) = Z_{TN} - \Delta$. (Inferior) Nos enfocamos en TN_3 y hacemos un acercamiento alrededor de éste. Se puede observar cómo las dos curvas de nivel establecen áreas blancas y grises sobre la cobertura de TN_3 .

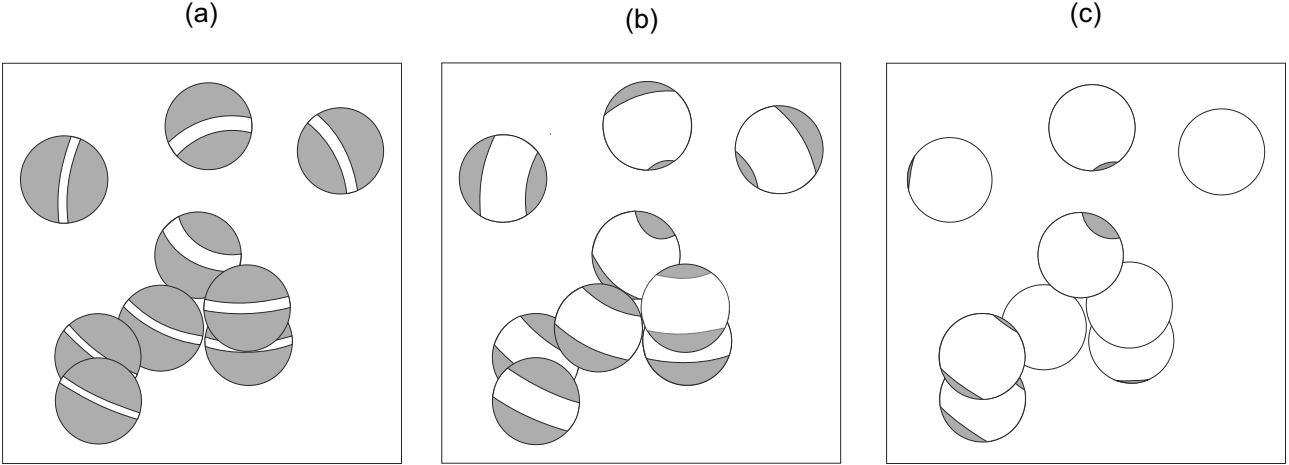


Figura 3.9: Áreas de desacuerdo (gris) para (a) $\Delta = 1$, (b) $\Delta = 5$, y (c) $\Delta = 10$

De nuevo, sólo los nodos ubicados en el área gris estarán en desacuerdo con el paquete de afirmación mientras que los nodos ubicados en las áreas blancas permanecerán en silencio. Como se puede observar en las figuras 3.9 (a,b,c), el área gris se hace más grande conforme el valor de la Δ disminuye, por lo cual es de esperarse que un valor pequeño en Δ reduzca el área en la cual un SN permanece en silencio.

Sea $A_{reply}^{TN_i}$ el área gris asociada a TN_i . Si m es el número de TN , y SN está en el alcance de transmisión de TN_i , incluyendo TN_0 ; la probabilidad de que un SN , ubicado en cualquier parte de la red permanezca en silencio, puede ser aproximada como:

$$Probability_{silent} \approx \frac{1}{mA_{coverage}^{TN_i}} \sum_{i=1}^m A_{Gray}^{TN_i} \quad (3.4)$$

En la figura 3.10 se muestra esta probabilidad para varios valores de Δ . En dicha figura se considera una red de sensores con 50 nodos, un alcance de transmisión de 10 metros y las lecturas entre los nodos vecinos se encuentran altamente correlacionadas.

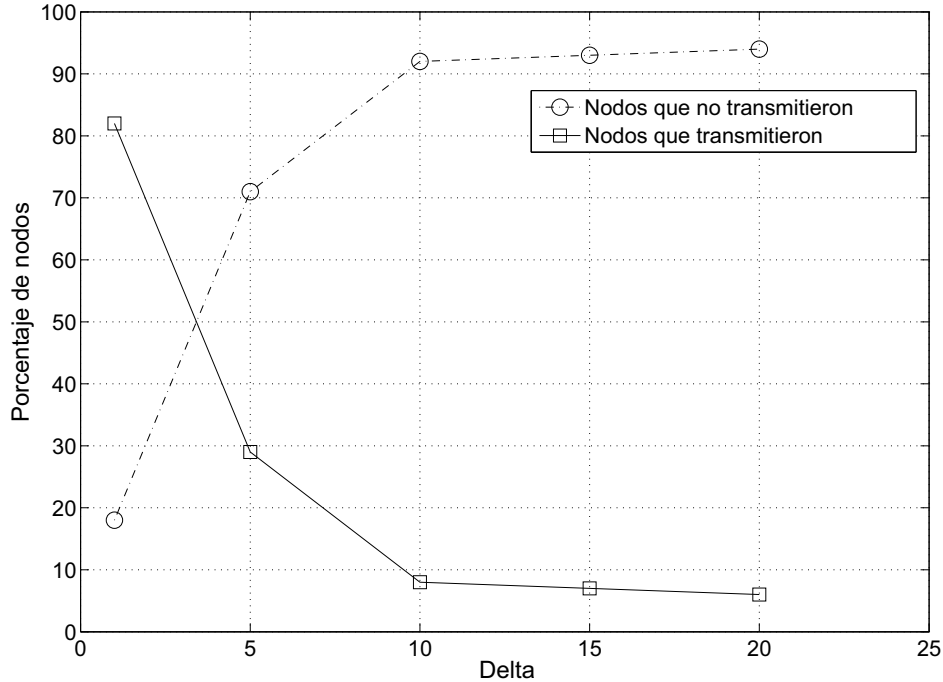


Figura 3.10: Porcentaje de nodos que transmitieron y los que no transmitieron

Como se puede observar en este ejemplo, aún para una Δ de 1, el 18% de los nodos permanecieron en silencio, mientras que el 92% permaneció en silencio para una Δ de 10 unidades. Por la forma de funcionar, en el esquema tradicional de *cluster-head* 100% de los nodos tuvieron que transmitir; al comparar los resultados de esta figura se puede observar fácilmente las ventajas que ofrece *Disagree* en términos de ahorro de energía. Finalmente, *Disagree* es un método que explota la correlación espacial de los datos, sin embargo, este método podría ser adaptado también para aprovechar la correlación temporal.

3.6 Implementación, simulaciones, experimentos y resultados

Implementamos *Disagree* en el simulador NS2.29. Para las etapas 1 y 2 se implementó el algoritmo de Prim [117] para construir un árbol mínimo utilizado para propagar las peticiones de información a nivel de capa de red. Las etapas 3 y 4 que comprenden el envío de paquetes de desacuerdo y el envío de las respuestas fueron implementadas en la capa de aplicación.

Para estudiar el desempeño de *Disagree*, se realizaron varios experimentos en distintos escenarios de monitoreo, contruidos con diferentes características. Se seleccionaron tres escenarios, denominados *Mosaic*, *Gaussian* y *Random*, mismos que presentan una correlación espacial distinta. Se utilizó una escala de grises para representar gráficamente las lecturas de 0 a 100 unidades, ver la parte superior de la figura 3.11. A continuación se describe cada uno de los escenarios mencionados:

Mosaic. Este escenario es una zona cuadriculada, compuesta de 100 cuadros fijos (10m por 10m cada uno), que cubre toda la red. Un intervalo fijo abarcando 10 unidades fue asignado aleatoriamente a cada cuadro. Por ejemplo, si a un cuadro se le asignó el intervalo de 70-80 unidades, entonces un nodo ubicado dentro de este cuadro, escogerá aleatoriamente valores entre 70 y 80 unidades. Vea la figura 3.11(a).

Gaussian. Para crear este escenario, se utilizó una distribución normal definida por la ecuación (3.5) e ilustrada en 3.11(b). En las simulaciones, la lectura del fenómeno que sería medida por un sensor se determina sustituyendo las coordenadas de su posición (x, y) en la ecuación (3.5). Se usó esta función sólo como ejemplo de un fenómeno donde las lecturas de nodos vecinos presentan variaciones suaves.

$$Reading(x, y) = 100e^{-(x-50)^2-(y-50)^2}*.0002 \quad (3.5)$$

Random. En este escenario, cada sensor selecciona aleatoriamente un valor entre 0 y 100 unidades, independientemente de otros sensores, vea la figura 3.11(c). Aquí, las lecturas

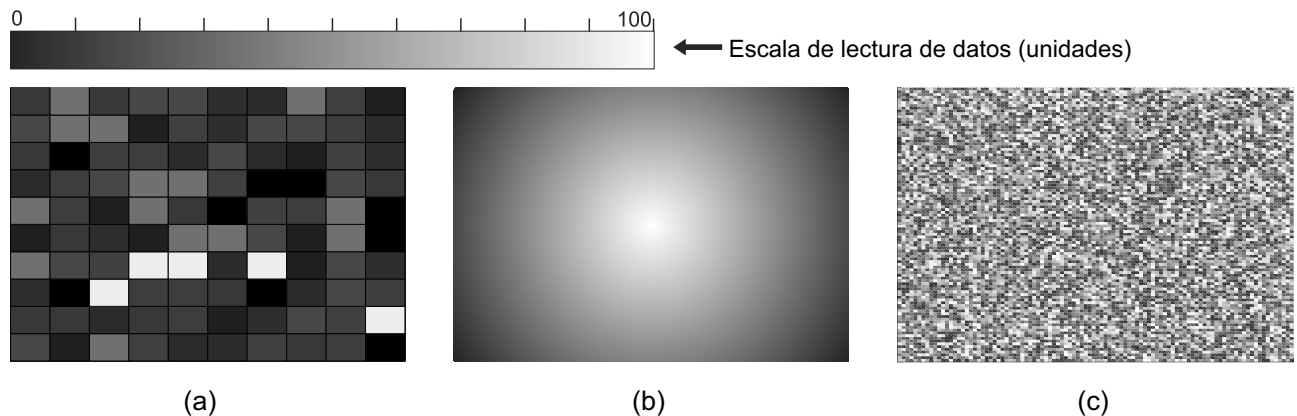


Figura 3.11: Escala de lectura de datos. Escenarios (a) Mosaic, (b) Gaussian y (c) Random

entre nodos vecinos no están relacionadas y las variaciones entre los datos de nodos vecinos pueden ser grandes.

3.6.1 Simulaciones

Para realizar las simulaciones se crearon diversas topologías con características diferentes, como la densidad de los nodos, el radio de propagación, la redundancia de los datos y los valores de Δ . Se consideraron 50, 100, 150, 200, 300, 400 y 500 nodos, el alcance de propagación varió entre 10 y 20 metros, los valores para Δ fueron 0, 10, 20, y 100 unidades.

Comparamos nuestros resultados con LEACH [30], solicitando una vista completa de la red, donde a todos los *SN* se les solicita que transmitan sus lecturas al *TN* con el que se encuentran asociados. Para LEACH, se utilizó la misma estructura de árbol que para *Disagree*. Adicionalmente, comparamos *Disagree* con CAG [31], usando valores *threshold* de 20 %, 10 % y 0 %. En los tres protocolos cada *cluster head* espera que los *cluster heads* que están asociados con él transmitan sus datos antes de transmitir toda la información recolectada hacia el nodo *sink*. Adicionalmente, el rol del *cluster head* es rotado en cada periodo de tiempo.

Para cada simulación se contabilizó el número de paquetes transmitidos, el número de nodos que permanecieron en silencio y el error de estimación durante todas las etapas de cada uno de los tres protocolos. Es importante mencionar que el concepto de Δ en *Disagree* es similar al concepto de error (*threshold*) en CAG, sin embargo no son directamente equivalentes. Por ejemplo, un *TN* en *Disagree* con *reading* = 90 y $\Delta = 20$ generan un intervalo de [70 – 110], mientras que en CAG, lo más aproximado sería un *cluster head* con *reading* = 90 y $\tau = 20\%$, dando un intervalo de [72 – 110]; en la Tabla 3.1 se muestran algunas equivalencias entre Δ (*Disagree*) y τ (CAG) para varias lecturas.

Las figuras 3.12, 3.13, 3.14 muestran el número de paquetes transmitidos en los escenarios *Gaussian*, *random* y *mosaic* durante todas las etapas de cada uno de los tres protocolos. Tomando en cuenta el desempeño de *Disagree*, se puede observar que el número de paquetes transmitidos se incrementa conforme el número de nodos también se incrementa para cualquier valor de Δ . Esto sucede porque al tener más nodos en la

Reading=20		Reading=50		Reading=80	
$\Delta(\text{Disagree})$	$\tau(\text{CAG})$	$\Delta(\text{Disagree})$	$\tau(\text{CAG})$	$\Delta(\text{Disagree})$	$\tau(\text{CAG})$
0	0 %	0	0 %	0	0 %
1	5 %	1	2 %	1	1.25 %
10	50 %	10	20 %	10	12.5 %
20	-	20	40 %	20	25 %
100	-	100	-	100	-

Tabla 3.1: Equivalencias entre $\Delta(\text{Disagree})$ y $\tau(\text{CAG})$

red, la probabilidad de transmisión de paquetes de desacuerdo también se ve incrementada. En estas figuras, el mayor número de paquetes transmitidos ocurre cuando $\Delta = 0$ unidades, lo cual es el resultado de que los nodos siempre transmitan sus lecturas a sus TN asociados, a menos de que las lecturas de los SN sean iguales a las de su TN . Por otra parte, cuando el valor de Δ se incrementa, disminuye el número de paquetes transmitidos debido a que más sensores permanecen en silencio. La principal diferencia entre estas tres figuras es que para el escenario *Gaussian*, las curvas para una Δ igual a 100, 20 y 10 unidades presentan un número similar de paquetes transmitidos en comparación con los otros dos escenarios, donde con una Δ de 20 y 10 unidades el número de paquetes transmitidos se incrementa notoriamente. Esta situación ocurre porque, mientras la mayoría de los sensores permanece en silencio en el escenario *Gaussian* para una $\Delta = 10$, pocos sensores envían sus paquetes de desacuerdo en el escenario *mosaic*, y casi todos los sensores estuvieron en desacuerdo en el escenario *random* con el mismo valor de Δ . En estas figuras se puede observar que CAG produce un comportamiento similar a *Disagree*.

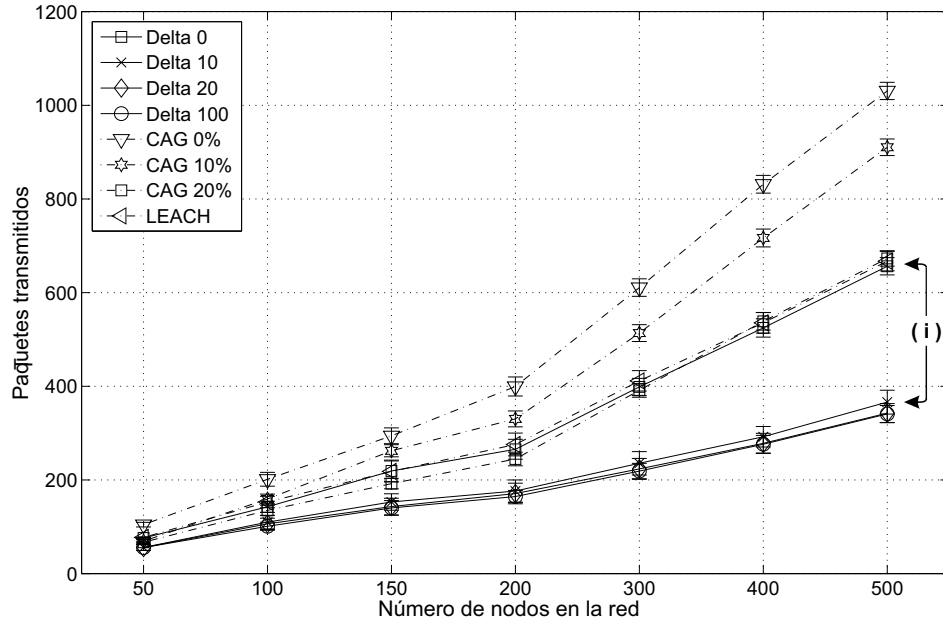


Figura 3.12: Paquetes transmitidos: Escenario *Gaussian* con intervalos de confianza del 95 %.

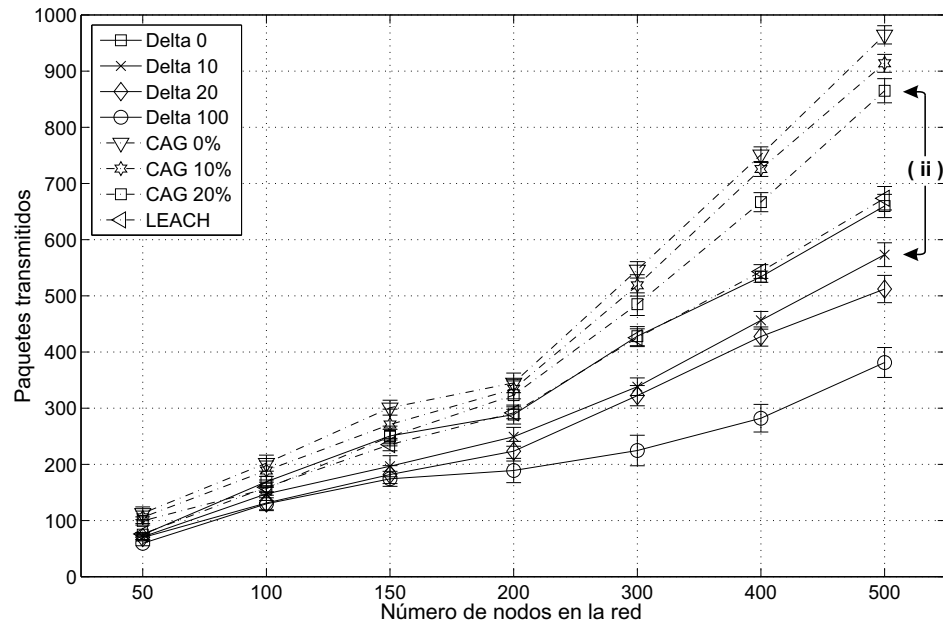


Figura 3.13: Paquetes transmitidos: Escenario *Mosaic* con intervalos de confianza del 95 %.

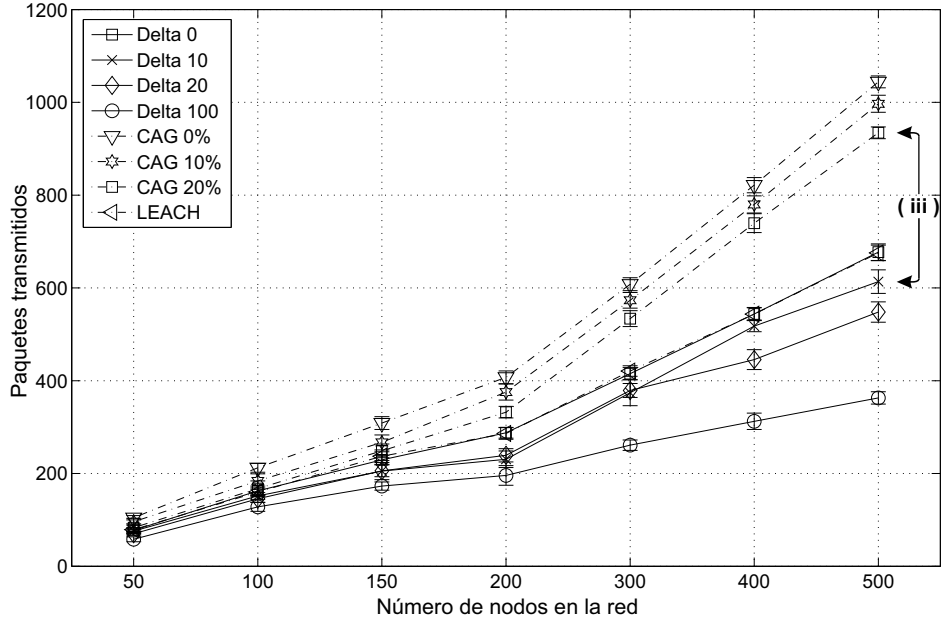


Figura 3.14: Paquetes transmitidos: Escenario *Random* con intervalos de confianza del 95 %.

Como se ya se mencionó, las curvas de *Disagree* y CAG no son directamente comparables, sin embargo, para facilitar una comparación se indican con una flecha (ver flechas i, ii e iii en las figuras 3.12, 3.13 y 3.14) tres curvas que se consideran de alguna manera equivalentes en términos de que producen la misma resolución espacial (ver tabla 3.1). Los resultados señalados por las flechas indican que CAG produce, en promedio, 24 % más de paquetes de señalización que *Disagree* en los tres escenarios. Este resultado se debe principalmente al hecho de que en *Disagree* hay más sensores que permanecen en silencio, y el árbol de distribución es menor en comparación con CAG. Finalmente, para el protocolo LEACH el número de paquetes transmitidos se encuentra entre las curvas de *Disagree* y CAG. Como se puede observar en estas figuras, el desempeño de *Disagree* supera el de LEACH excepto para un $\Delta = 0$, donde ambos protocolos producen resultados similares. Esto ocurre debido a que en *Disagree* hay nodos que permanecen en silencio, mientras que en LEACH todos los nodos siempre transmiten. Para lograr un ahorro de energía, se recomienda usar *Disagree* con $\Delta > 0$. Por otro lado, el desempeño de LEACH es mejor que CAG para cualquier valor de τ , esto ocurre porque LEACH tiene un árbol de distribución menor. Es importante mencionar que, en la implementación de LEACH se utilizó un árbol mínimo para la distribución de los nodos. En caso de que el

árbol de distribución sea distinto de un árbol mínimo, LEACH producirá más paquetes de señalización que los presentados en estas figuras. Las figuras 3.15, 3.16 y 3.17 muestran el porcentaje de nodos que permanecieron en silencio considerando los tres escenarios. Como es de esperarse, en *Disagree* más sensores permanecen en silencio conforme el valor de la Δ se incrementa y estos porcentajes no cambian mucho aunque varíe el número de nodos en la red. Mientras que estas figuras exhiben tendencias similares, la diferencia más importante ocurre, una vez más, en el escenario *Gaussian* pues cerca del 50 % de los nodos permanecieron en silencio para una Δ con valores de 10, 20 y 100 unidades, mientras que los otros dos escenarios, con una Δ de 20, presentan 30 % y 20 % de nodos en silencio para los escenarios *mosaic* y *random* respectivamente. Para CAG se observa que, para una resolución espacial equivalente (ver los valores señalados con las flecha i, ii, y iii), hay 20 % menos nodos en silencio que en *Disagree*. A pesar de que en CAG hay menos *cluster heads*, los nodos intermedios deben reenviar los paquetes, ocasionando que el número de nodos en silencio en CAG sea menor que en *Disagree*. Finalmente, para LEACH se puede observar que el porcentaje de nodos en silencio siempre es 0 % debido a que todos los nodos son requeridos para transmitir.

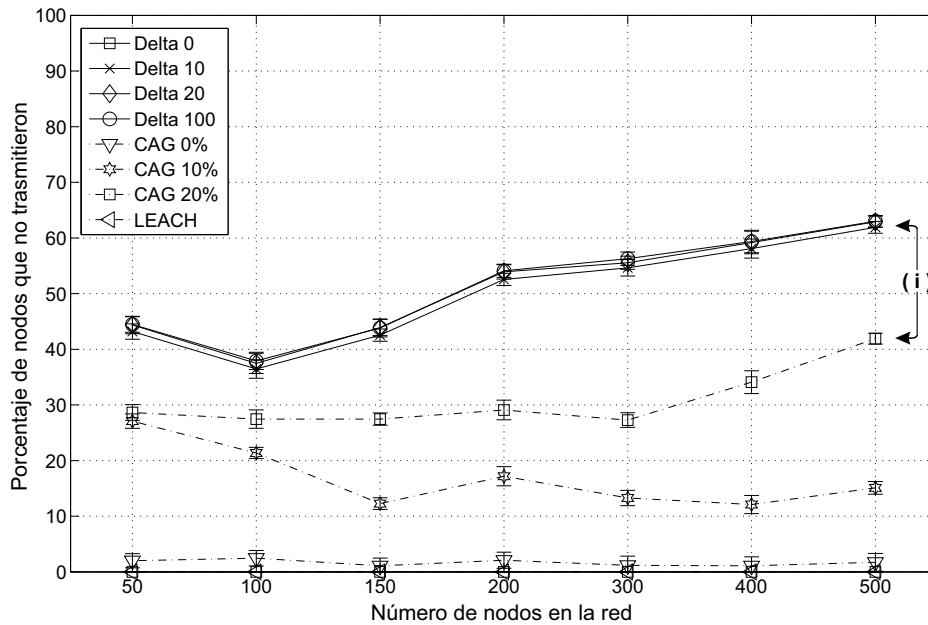


Figura 3.15: Porcentaje de nodos que no transmitieron: Escenario *Gaussian* con intervalos de confianza del 95 %.

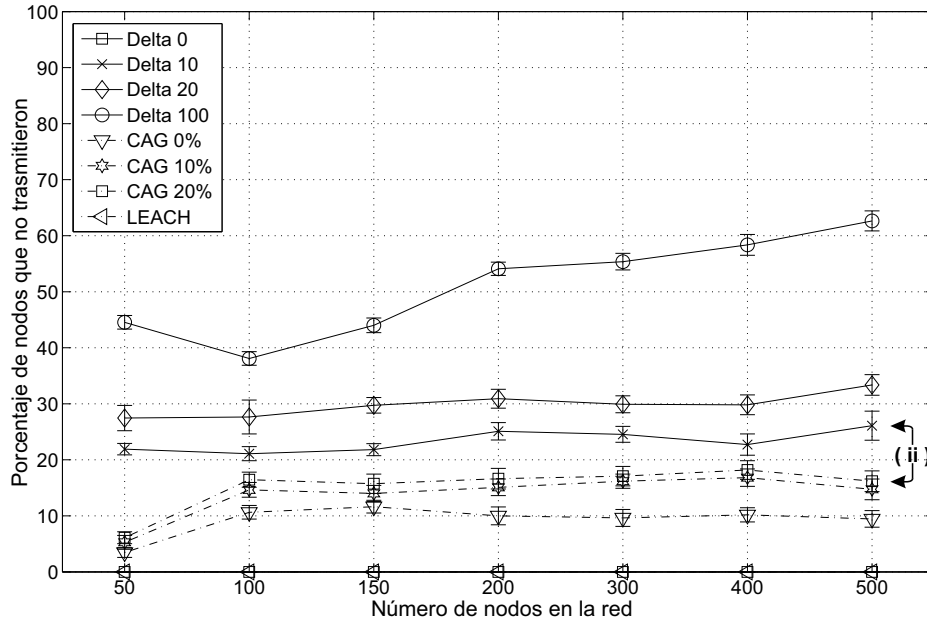


Figura 3.16: Porcentaje de nodos que no transmitieron: Escenario *Mosaic* con intervalos de confianza del 95 %.

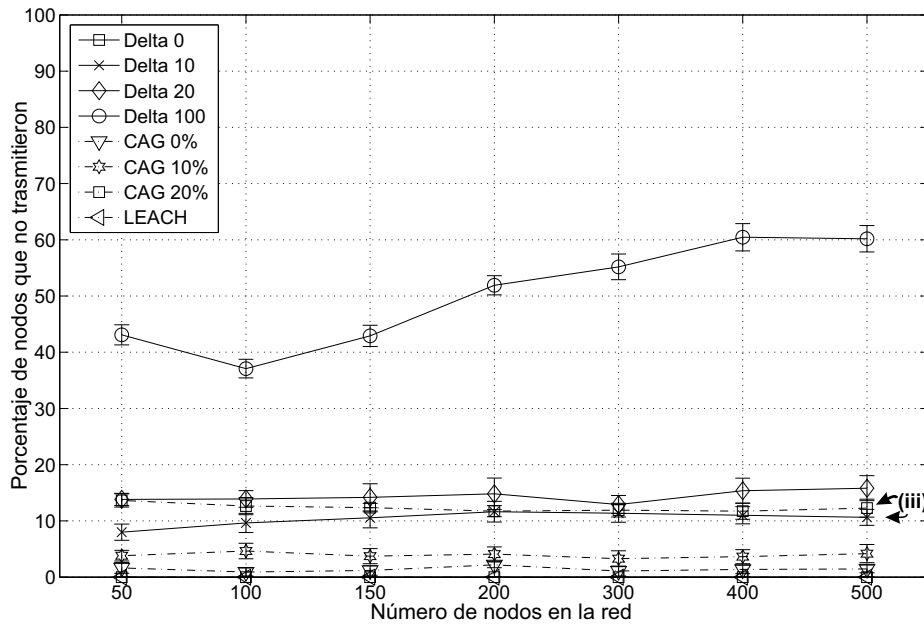


Figura 3.17: Porcentaje de nodos que no transmitieron: Escenario *Random* con intervalos de confianza del 95 %.

Las figuras 3.18, 3.19 y 3.20 muestran el error absoluto. Este error de estimación es la diferencia absoluta entre las lecturas reales y las estimadas de cada sensor. Como se puede observar, para valores de Δ mayores a las 10 unidades, el error de estimación es bajo (dentro de 5 unidades) para los tres escenarios, sólo para una $\Delta = 100$, el error aumenta hasta 20 unidades para los escenarios *mosaic* y *random*. El mismo comportamiento puede observarse en CAG, pero una vez más, se aprecia que *Disagree* alcanza errores de estimación menores en comparación con CAG para una resolución espacial similar. En LEACH no hay error de estimación porque todos los nodos transmiten sus lecturas exactas. La resolución espacial equivalente es indicada con las flechas i, ii y iii.

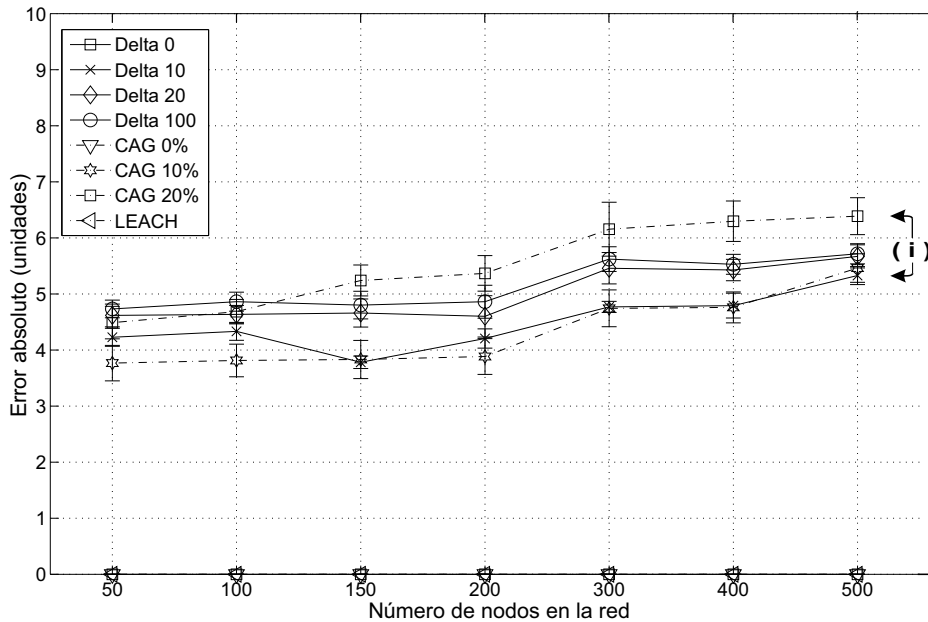


Figura 3.18: Error de estimación: Escenario *Gaussian* con intervalos de confianza del 95 %.

Resumiendo el comportamiento observado en todas las figuras, se puede apreciar que por un lado en el escenario *Gaussian* hay pocos paquetes de desacuerdo combinado con pequeños errores de estimación, por otro lado, en el escenario *random*, para cualquier valor de Δ , hay muchos paquetes de desacuerdo y el error de estimación es alto. Más importante aún, para el escenario *random* el desempeño de *Disagree* es mejor que el del método LEACH, en el cual todos los nodos reportan sus lecturas a sus *cluster heads* asociados.

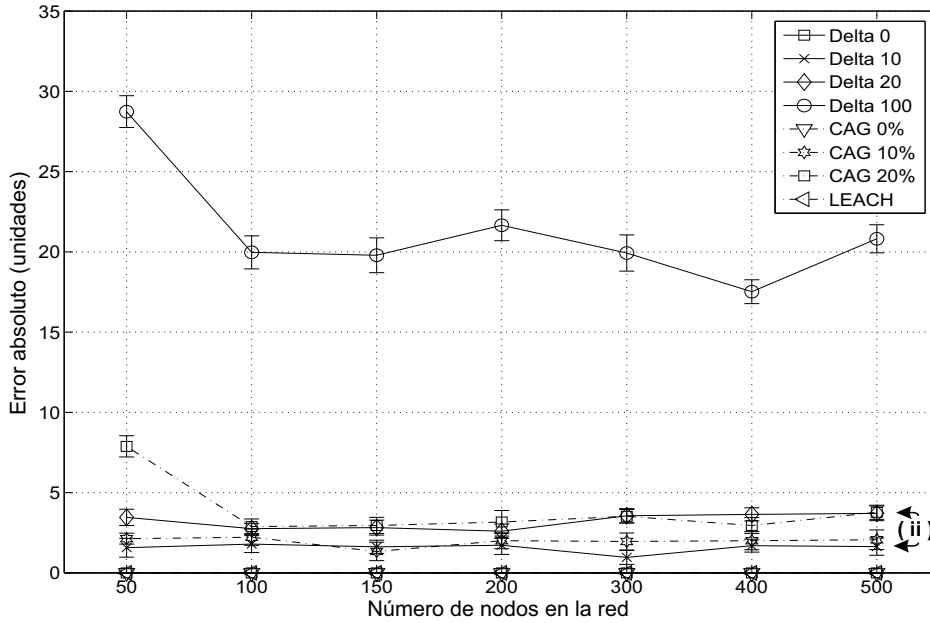


Figura 3.19: Error de estimación: Escenario *Mosaic* con intervalos de confianza del 95 %.

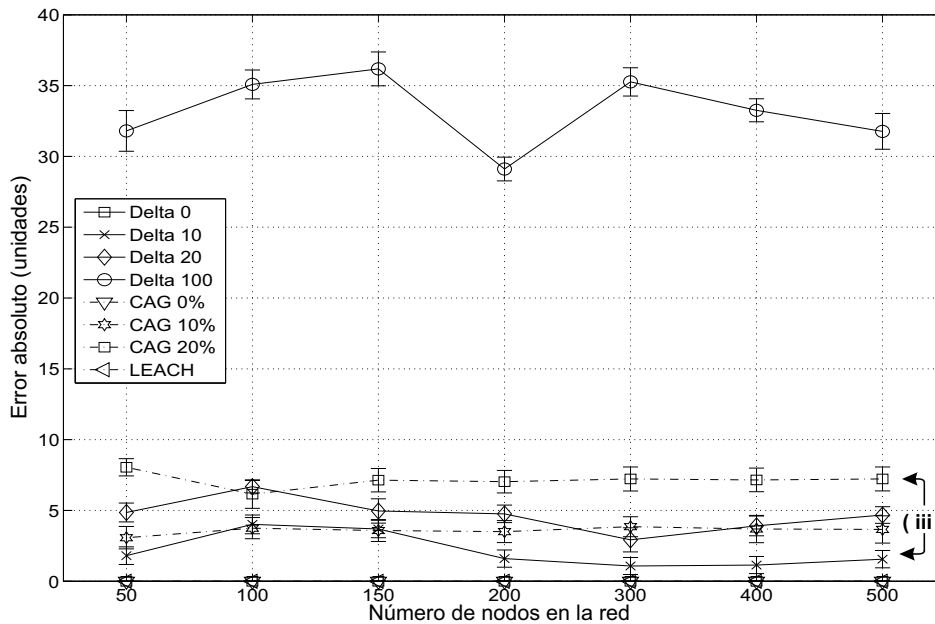


Figura 3.20: Error de estimación: Escenario *Random* con intervalos de confianza del 95 %.

El alcance de transmisión impacta el comportamiento de *Disagree* en dos formas. Si se incrementa dicho alcance se reduce el número de *TN* en el árbol pero se incrementa el

número de SN asociados por TN . De manera similar, un alcance de transmisión muy grande incrementa la probabilidad de que SN asociados y ubicados más lejos estén en desacuerdo con el paquete de afirmación debido a que la correlación espacial disminuye con la distancia. En la figura 3.21 se muestra cómo en *Disagree* el alcance de transmisión afecta el número de paquetes transmitidos; se consideraron valores de 10 y 20 metros, respectivamente. En general, para cualquier escenario las curvas para $R = 20m$ estuvieron por debajo de la curva equivalente para $R = 10m$. Esto es porque para $R = 20m$ el árbol es más pequeño (ver la figura 3.22), y también porque el número de SN que reportan a los TN se incrementa, nodos que potencialmente pueden permanecer en silencio para valores grandes de Δ .

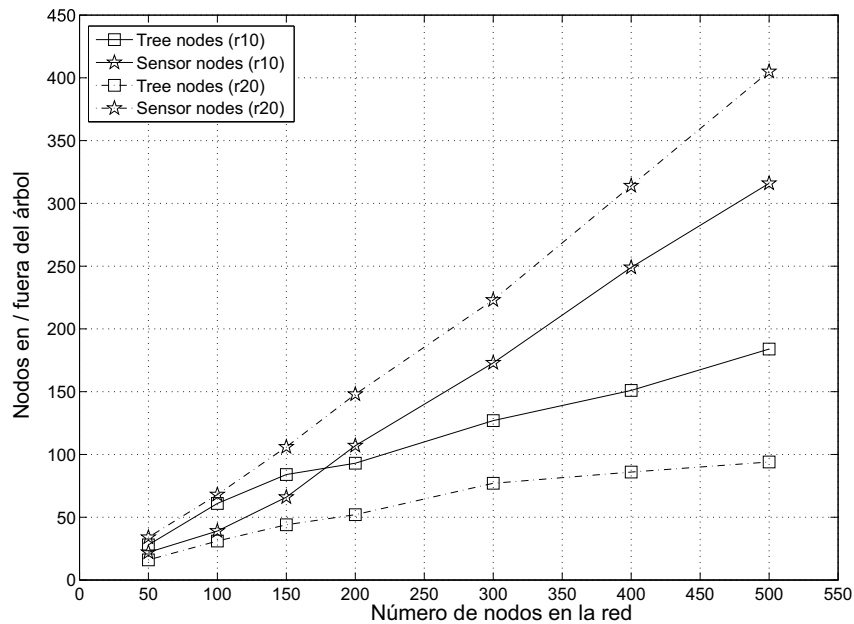


Figura 3.21: Número de TN (nodos árboles) y SN (nodos sensores)

Finalmente, la figura 3.23 muestra el número de veces, durante un intervalo de 10 épocas, en que cada nodo fue parte del árbol en una red de 200 nodos. Se puede observar que la mayoría de los nodos (78%) fueron TN en 5 ocasiones. Algunos nodos (19%) nunca fueron TN porque dichos nodos se encuentran ubicados en las fronteras de la red. El nodo *sink* siempre formó parte del árbol porque no fue incluido en el proceso de rotación. Como puede observarse en la Figura 3.23, el papel de nodo árbol está distribuido homogéneamente entre los nodos.

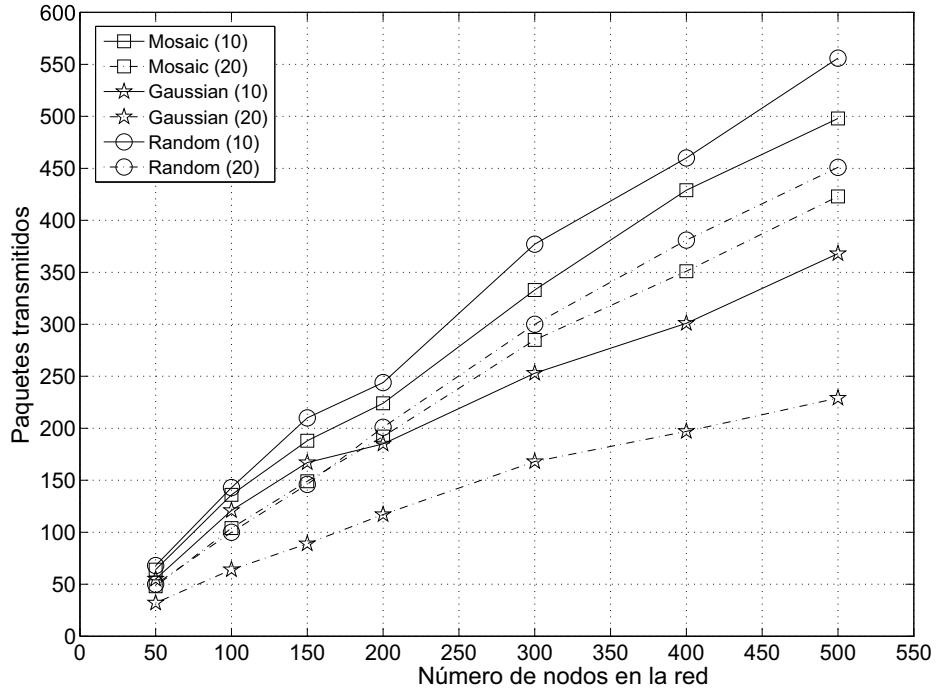


Figura 3.22: Comparación de escenarios

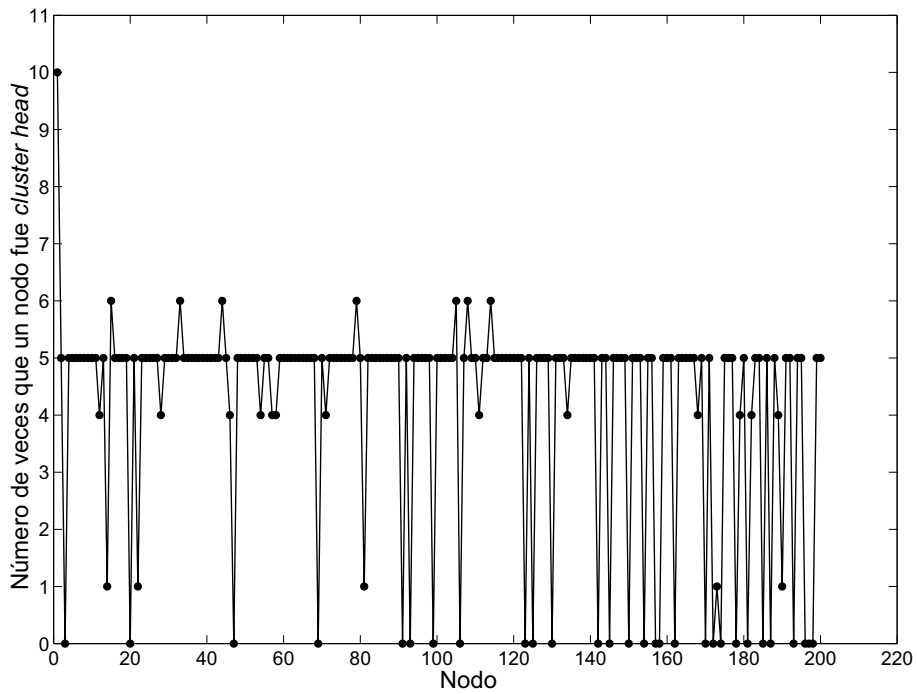


Figura 3.23: Rotación del rol del nodo árbol

Para medir el consumo de energía se utilizó un modelo similar al de [30], en donde se establece que para transmitir un mensaje de l bits a una distancia d , el transmisor (*elec*) y el amplificador (*amp*) gastan energía en hacer funcionar el radio, dicho gasto puede ser representando mediante la expresión 3.6

$$E_{Tx}(l, d) = E_{Tx-elec}(l) + E_{Tx-amp}(l, d) \quad (3.6)$$

Debido a que la atenuación de la potencia es dependiente de la distancia (d_0) que existe entre el transmisor y el receptor, se utilizaron modelos de propagación diferentes de tal forma que si la distancia es menor que $d_0 = 86m$, se seleccionó el modelo de propagación *free space* ϵ_{fs} y si la distancia es mayor a d_0 , se utilizó el modelo *multipath fading* ϵ_{mp} , tal como se expresa en 3.7

$$E_{Tx}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{fs-amp}d^2, & d < d_0 \\ lE_{elec} + l\epsilon_{mp-amp}d^4, & d \geq d_0 \end{cases} \quad (3.7)$$

y para recibir un mensaje, el nodo gasta:

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} \quad (3.8)$$

Para las simulaciones y los experimentos descritos en este trabajo, los parámetros utilizados para el consumo de energía son similares a [30] y se listan a continuación:

$$E_{elec} = 50nJ/bit \quad (3.9)$$

$$\epsilon_{fs-amp} = 10(pJ/bit)/m^2 \quad (3.10)$$

$$\epsilon_{mp-amp} = 0.0013(pJ/bit)/m^4 \quad (3.11)$$

Para comparar *Disagree* con CAG y LEACH en términos de eficiencia en energía, simulamos los tres protocolos durante 20000 eventos. Para cada evento, el nodo *sink* solicita una visión completa de la red y los sensores transmiten sus lecturas. El papel de *cluster head* se rota después de cada 10 eventos. De manera similar a LEACH, se corrieron simulaciones hasta que a todos los nodos se les agotó la batería. Los resultados obtenidos se

muestran en la figura 3.24, donde puede observarse que el tiempo de vida de la red para *Disagree* es mayor comparado con CAG o LEACH. CAG presenta el tiempo de vida de red más corto porque la rotación del *cluster head* necesariamente involucra el cambio de topología, lo cual requiere que todos los nodos reenvíen paquetes para escoger los nuevos nodos tipo *cluster*, *puente* y no-participantes.

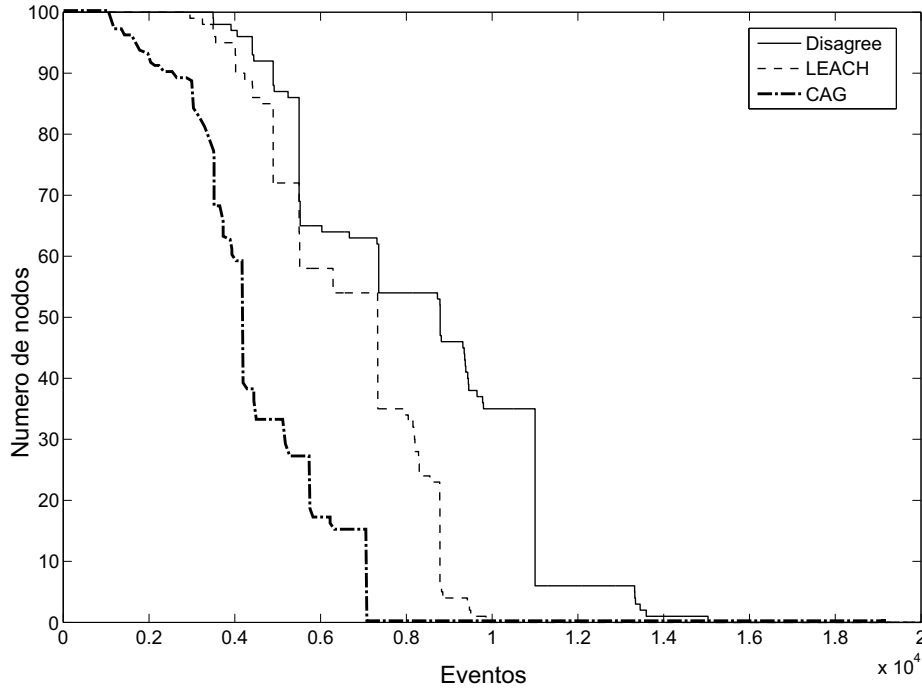


Figura 3.24: Número de nodos vivos

3.6.2 Experimentos

Implementamos *Disagree* en dos plataformas: *tmote sky* y *telosb*. La figura 3.25 ilustra la ubicación de los nodos, el árbol resultante utilizado para las etapas de expansión y respuesta de *Disagree* y también las lecturas en cada nodo. Dichas lecturas no cambiaron durante los experimentos y el valor de Δ utilizado fue 10.

El objetivo de estos experimentos es comparar el consumo de energía real en los nodos tipo árbol y tipo sensor. El nodo *sink* solicitó una visión completa de la red cada 10 segundos. Enfocándonos en TN_1 , tiene dos *SNs* asociados: SN_A nunca estuvo en desacuerdo mientras que SN_B siempre estuvo en desacuerdo. En la tabla 3.2, se muestra

el tiempo de vida de los nodos asociados a TN_1 . En esta tabla se puede observar que el tiempo de vida de SN_A duró 30% más comparado con SN_B . Esto claramente muestra el potencial que tiene *Disagree* para el ahorro de energía. Esta tabla también muestra el tiempo de vida de los TN , enfatizando la necesidad de rotar el rol de los nodos tipo árbol periódicamente para evitar que se les agote la batería prematuramente.

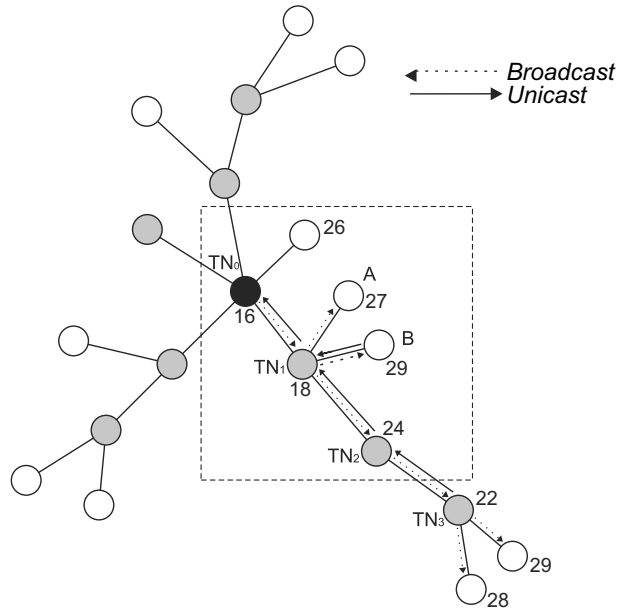


Figura 3.25: Experimentos

Nodo	Tiempo de vida (h)
TN_0	140
TN_1	152
TN_2	115
SN_A	215
SN_B	147

Tabla 3.2: El tiempo de vida de los nodos

3.7 Conclusiones del capítulo

En este capítulo se propone *Disagree*, un nuevo método para la adquisición de datos en las redes inalámbricas de sensores estáticos que explota la correlación espacial de datos al nivel del sensor con el fin de ahorrar energía. Más que solicitar a todos los sensores el envío de información para obtener una visión completa de la red, *Disagree* solicita información a aquellos sensores cuyas lecturas estén fuera de un intervalo centrado en la lectura reportada por el nodo árbol con el que está asociado el nodo sensor. En este capítulo se demuestra que este comportamiento es la esencia de una técnica eficiente en cuestión de ahorro energía para la recolección de información en la red de sensores. Las lecturas de los nodos que no transmiten pueden ser estimadas fácilmente, y el error de estimación puede ser controlado a través de la correcta definición del intervalo a considerar, el cual se establece a través de un parámetro.

Disagree fue implementado en el simulador NS2.29 y se llevaron a cabo varios experimentos para probar sus ventajas y desventajas en escenarios con distintas características. Los resultados indican que el desempeño mejora (pocos paquetes transmitidos) conforme las variaciones entre las lecturas de los nodos vecinos se hacen más pequeñas, lo cual es una propiedad común de muchos fenómenos físicos. En el peor caso, *Disagree* se desempeña de la misma forma que el esquema tradicional basado en *cluster head*, en el cual todos los sensores responden a las solicitudes de información cuando una visión completa de la red es requerida.

Capítulo 4

Método *Urbihoc*

4.1 Introducción

En este capítulo se presenta *Urbihoc*, el cual es un método de recolección de datos que aprovecha la movilidad de los sensores para monitorear fenómenos en áreas grandes, como por ejemplo una ciudad. Cada nodo móvil que integra la red WSN puede adquirir una vista completa de lo que sucede con el fenómeno de interés, a través de sus lecturas más los intercambios locales realizados con otros nodos.

Diariamente la gente se desplaza a diferentes lugares utilizando su automóvil o unidades de transporte público y llevando consigo sus teléfonos móviles. El automóvil, la unidad de transporte público y el teléfono celular pueden tener integrados sensores para adquirir información acerca del comportamiento de ciertos fenómenos [27], tales como la contaminación, el ruido, el tráfico, el estado de las vialidades, etc. A este tipo de monitoreo se le conoce como monitoreo urbano [125, 126] (*Urban sensing*). Los sensores que pueden ser integrados a los automóviles y a las unidades de transporte pueden ser más especializados que aquellos que están integrados a los teléfonos celulares (*smartphones*), sin embargo pueden trabajar en conjunto integrando una red WSN de nodos móviles para compartir información. Dichos nodos móviles podrían obtener una vista completa de lo que sucede con el fenómeno de interés si compartieran su información adquirida, ya sea enviando los datos recolectados a un servidor central a través de la red celular, o de servicios WiFi disponibles, o enviándolos directamente a sus nodos vecinos. La red celular ofrece buena cobertura y acceso continuo, sin embargo la capacidad de cada celda

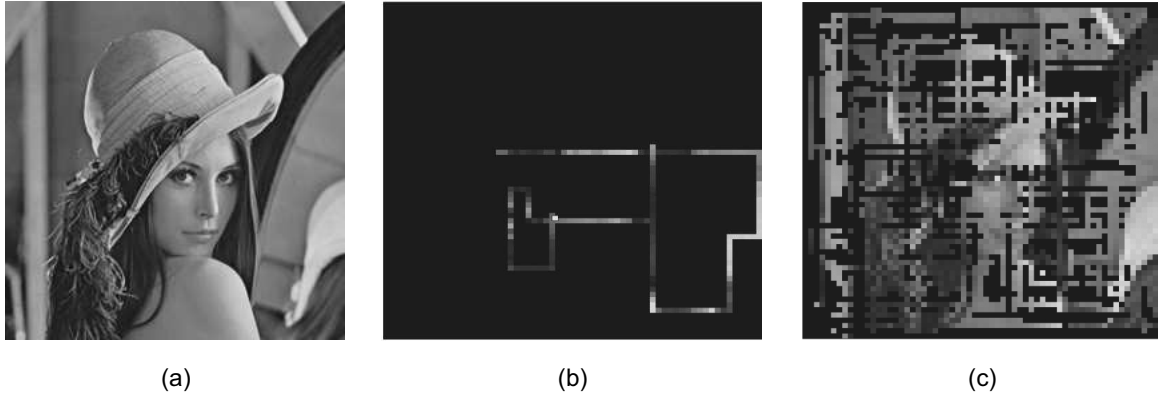


Figura 4.1: (a) Imagen que representa el fenómeno a monitorear. (b) Datos adquiridos por un nodo móvil durante una jornada. (c) Datos adquiridos por el mismo nodo móvil gracias a los intercambios realizados con otros nodos móviles utilizando WiFi en modo ad hoc.

está limitada y este servicio requiere la implementación de una costosa infraestructura. Los servicios de WiFi no tienen una cobertura tan extensa y la implementación y acceso a ellos también resultan costosos. Para lograr que cada nodo móvil adquiriera una vista completa del fenómeno sin necesitar de infraestructura costosa se propone *Urbihoc*, el cual es un método para monitorear, recolectar y compartir información utilizando sensores móviles. En este método el envío de información se lleva a cabo utilizando sólo WiFi en modo ad hoc. El objetivo principal de *Urbihoc* es que cada nodo móvil adquiriera una vista completa del fenómeno de interés a partir de intercambios locales con otros nodos móviles exclusivamente.

Para explicar gráficamente los beneficios de *Urbihoc*, se utilizan las imágenes que se muestran en la figura 4.1. La figura 4.1(a) representa el fenómeno que se quiere monitorear, en esta figura cada pixel representa el valor del fenómeno monitoreado en determinada ubicación ¹. En este escenario, cada nodo móvil viaja diariamente tomando lecturas del fenómeno y las guarda en su memoria. Los datos que dicho nodo móvil adquirió durante cierto periodo se muestran en la figura 4.1(b). A lo largo de su viaje, el nodo móvil se encontró dentro del alcance de transmisión de otros sensores móviles e intercambió datos con ellos utilizando su interfaz WiFi en modo ad hoc. La figura 4.1(c) muestra toda la

¹Se seleccionó la imagen de la figura 4.1(a) sólo para simular el fenómeno o el escenario, sin embargo cualquier otra imagen puede ser utilizada.

información “aprendida” por un nodo móvil más aquella información que adquirió gracias a los intercambios realizados con otros nodos. Comparando la información adquirida en las figuras 4.1(b) y 4.1(c), se pueden observar los beneficios de compartir información con los nodos cercanos ya que cada nodo móvil puede tener acceso a más información, en especial, de regiones que todavía no han visitado durante su recorrido. También puede observarse en la figura 4.1(c), que los datos adquiridos lucen similares a los datos mostrados en la figura 4.1(a), lo cual significa que el nodo pudo adquirir una buena aproximación de la vista completa del fenómeno monitoreado.

4.2 Formas de recolección de datos

Las aplicaciones que realizan monitoreo urbano utilizando nodos móviles pueden recolectar información de dos maneras: *participativa* [127, 128] y *oportunistista* [129]. En el monitoreo participativo un usuario decide conscientemente responder o colaborar con alguna requisición de información que le haya llegado, recolectando y compartiendo los datos obtenidos con otros usuarios. En el monitoreo oportunista, la recolección y el compartir de los datos son actividades completamente automatizadas por lo que no se requiere la intervención del usuario. *Urbihoc* opera en modo oportunista, por lo que no es necesario que el usuario esté pendiente de los procesos que realiza la aplicación. Utilizar dispositivos portátiles equipados con sensores para el monitoreo de ciertos fenómenos aprovechando el poder de las multitudes, ha dado origen a una forma especial de monitoreo que algunos autores [89, 130] han denominado monitoreo basado en multitudes (*crowdsensing*). *Urbihoc* está relacionado con esta forma de monitoreo pues entre mayor sea el número de nodos móviles adquiriendo e intercambiando información, mayor es la probabilidad de obtener una vista completa acerca del fenómeno.

Al utilizar *Urbihoc*, el detalle de la vista completa del fenómeno de interés que adquiere un nodo móvil, dependerá del número de sitios o zonas visitadas y del número de otros nodos móviles con los cuales se haya encontrado a lo largo de sus trayectorias. Adquirir una vista completa puede tomar tiempo, por ello, *Urbihoc* es adecuado en aplicaciones cuyo monitoreo permita cierto retardo en la obtención de información tales como la contaminación, la temperatura, el nivel de ciertas sustancias en el ambiente, etc. En *Urbihoc*, cada nodo móvil almacena los datos adquiridos y sólo comparte su información cuando

se encuentra dentro de la cobertura WiFi de otro nodo móvil que está ejecutando la aplicación. Aquí el encaminamiento no es el objetivo, el objetivo es adquirir información y compartirla de manera oportunista con otros nodos para obtener una vista completa de lo que sucede con el fenómeno. *Urbihoc* no es una solución de salto múltiple pues no utiliza nodos intermedios para retransmitir (*relaying nodes*) datos. La estrategia consiste en transmitir datos de la fuente al destino (*single-hop*) inmediatamente después de que dos o más nodos se encuentren mutuamente dentro de su alcance de transmisión.

Como ya se mencionó, en *Urbihoc* los nodos móviles pueden ser vehículos (particulares o de transporte público), teléfonos celulares o algún dispositivo que tenga incluido algún tipo de sensor y una interfaz WiFi. En los vehículos es posible instalar sensores más especializados que en los teléfonos celulares, además de que no están tan limitados en energía. En *Urbihoc* se puede construir una red WSN con nodos móviles de la misma categoría es decir, sólo vehículos o sólo teléfonos, etc., pero también se puede construir una solución mixta de nodos móviles para aprovechar las ventajas de cada categoría. Por ejemplo, si lo que se requiere monitorear son los niveles de ciertas partículas en el aire, los sensores especializados pueden instalarse en los vehículos y los teléfonos celulares podrían encargarse de compartir la información con otros nodos móviles en diversos lugares, sobretodo aquellos lugares a los que los vehículos no pueden ingresar tales como el transporte público subterráneo, calles cerradas, dentro de edificios, hospitales, etc.

4.3 Retos

Obtener una vista completa del fenómeno de interés utilizando una red de nodos móviles organizados en modo ad hoc hace frente a los siguientes retos:

- La duración del encuentro o el contacto entre los nodos móviles es desconocida, pues depende de la velocidad, el alcance de transmisión y las trayectorias sobre las que se desplazan. Se requiere de un método para intercambiar datos capaz de transmitir la mayor cantidad de información posible durante un encuentro corto, gastando el mínimo de energía. Por lo tanto, la potencia y la velocidad de transmisión así como los esquemas de modulación y otros factores influyen en este aspecto.

- El espacio de almacenamiento en algunos nodos móviles es muy limitado por lo que sólo se podrá almacenar cierta cantidad de información, que dependerá de la frecuencia y la precisión con la que se tomen las lecturas del fenómeno y el tamaño en bytes de cada registro. Por ejemplo, en la figura 4.2 se ilustra el número de bytes que se requieren para diferentes frecuencias de monitoreo. Si el fenómeno se monitorea cada milisegundo y cada lectura ocupa 70 bytes y se requiere tener 2 horas de información, el teléfono debe almacenar 504 MB. En caso de que el teléfono sólo tenga 2 GB de almacenamiento disponible, resulta poco viable que se ocupe la cuarta parte para una sola aplicación. En cambio, si el monitoreo se lleva a cabo cada segundo tan sólo se requiere espacio para almacenar 504 kB.
- El tiempo de vida de los datos puede ser corto o largo dependiendo de la naturaleza del fenómeno a monitorear, es decir, si en la ubicación (x_0, y_0) se monitorea el fenómeno en el tiempo t_0 , y se tiene un valor z_0 , quizá éste valor ya sea obsoleto 30 minutos después porque ese mismo punto (x_0, y_0) , en el tiempo t_{30} tiene un nuevo valor z_{30} que es muy distinto de z_0 . *Urbihoc* emplea un factor de olvido para desechar la información obsoleta y mantener una ventana de información actualizada.
- Utilizar WiFi para realizar los intercambios locales de datos entre nodos cercanos, puede resultar muy costoso en términos de energía. En el caso de que los nodos móviles sean vehículos, el impacto de dicho gasto es menor debido a que los sensores pueden estar conectados a la batería de los vehículos. Sin embargo, para el caso en el que los nodos móviles sean teléfonos celulares, el impacto del gasto es mayor aunque estos dispositivos móviles son recargados por sus propietarios de forma rutinaria. Para minimizar este problema, *Urbihoc* puede "mandar dormir" a los nodos para que ahorren energía, esto ocasionará que se pierdan de compartir y que les compartan información por determinado periodo pero lograrán ahorrar energía.

En este trabajo se considera que *Urbihoc* funciona con interfaces WiFi, ya que éstas se encuentran disponibles en muchos de los equipos móviles actuales. El método podría operar con otras tecnologías de comunicación inalámbrica (por ejemplo *ZigBee* [131]), siempre y cuando éstas permitan la transmisión de información en forma directa, sin uso de infraestructura o comunicaciones multisalto. Para poder utilizar *Urbihoc* con otro tipo de interfaces inalámbricas habría que tomar en consideración el alcance de transmisión que éstas tengan ya que esto limita el tiempo de los encuentros entre los nodos móviles.

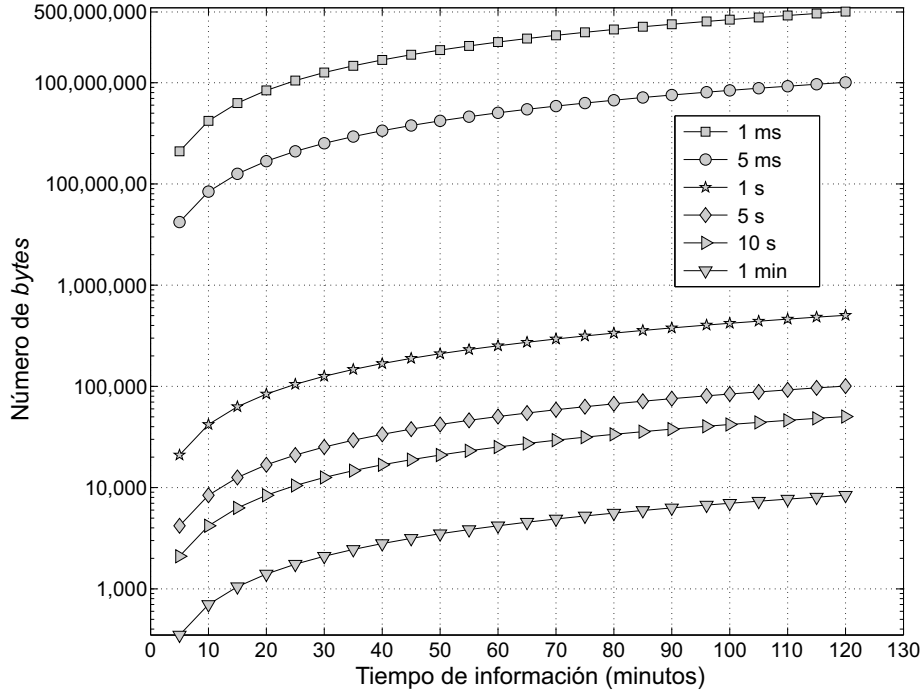


Figura 4.2: Frecuencia de monitoreo del fenómeno de interés y el tiempo de información

4.4 Trabajo relacionado

Existe un número importante de aplicaciones emergentes que usan nodos móviles para el monitoreo de ciertos fenómenos. Basándonos en su operación hemos clasificado estas aplicaciones en tres grandes grupos:

En el *primer grupo*, los nodos móviles monitorean periódicamente un fenómeno y envían los datos adquiridos hacia un servidor de datos central a través de la red celular o de servicios WiFi. Por ejemplo, *Nericell* [25] utiliza teléfonos móviles para detectar baches, golpeteo, frenados abruptos, etc. En [94] los autores utilizan los datos del acelerómetro para clasificar los movimientos físicos del usuario que carga el teléfono, y se comparte esta información a través de redes sociales como *Facebook*[132]. En [133] los autores desarrollaron un sistema que puede inferir si el usuario del teléfono móvil está caminando, anda en bicicleta o está tomando el autobús. Los datos adquiridos por este sistema son enviados a un servidor central para compartirlos con otros usuarios. En [134] los autores utilizan nodos móviles (teléfonos celulares y vehículos) para recolectar lecturas de los

niveles de ruido, las cuales son enviadas hacia un servidor de datos para construir mapas de ruido. En *Pothole Patrol* [24] los autores utilizan vehículos equipados con varios sensores para monitorear las condiciones de las carreteras. En esta solución los datos adquiridos son enviados a un servidor central utilizando conexiones WiFi oportunistas o la red celular si está disponible. Los autores implementaron un buffer local que permite almacenar temporalmente la información mientras ésta es enviada con cierto retardo. Otras soluciones en esta categoría son [127, 135, 136, 137].

Adicionalmente a que los nodos móviles envíen y soliciten información a un servidor central, en algunas soluciones correspondientes al *primer grupo*, el servidor es quien envía solicitudes de información a los nodos, y sólo aquellos que satisfagan la solicitud recibida enviarán sus datos al servidor que les realizó dicha solicitud. Por ejemplo, en [138] los autores proponen una aplicación que asocia los datos adquiridos con el tiempo y la ubicación del dispositivo, creando un registro llamado *microblog*. Este *microblog* es transmitido hacia un servidor utilizando la red celular o WiFi. En esta solución los usuarios pueden solicitar datos al servidor indicando el área geográfica de interés; si el servidor no tiene información disponible envía dicha solicitud a los nodos móviles que estén cerca de dicha área para obtener datos. En [139] los autores explotan de manera oportunista la colaboración de los peatones de la siguiente forma: una vez que los nodos móviles han recibido la solicitud del servidor, cada nodo revisa periódicamente si se encuentra dentro del área de la cual se está pidiendo información. Si se cumple la anterior condición, el nodo móvil comienza a monitorear el fenómeno y envía los datos adquiridos hacia el servidor. El nodo móvil reduce la tasa de transmisión a la que está subiendo información al servidor si éste detecta que hay otro nodo móvil cercano, pues ahora serán dos nodos los que estarán enviando información al servidor. Otra aplicación en este *primer grupo* es [140], en la cual se utilizan instrumentos para monitorear la radiación solar en la superficie. Estos instrumentos bombean aire a través de un filtro que no proporciona lecturas precisas o granulares. Para mejorar la calidad de los datos que se recolectan, los autores utilizan teléfonos móviles para tomar fotos de los filtros y estas imágenes son enviadas a un servidor central. En [137], los autores presentan tres aplicaciones: *citizen journalist* en la cual un usuario solicita tomar fotos de algún lugar y transmitir las a un servidor; en *party thermometer* las solicitudes son enviadas a los usuarios que estén en alguna fiesta, el usuario que recibe la solicitud toma muestras del sonido y las envía, jun-

to con su ubicación, a un servidor central; finalmente, en *road bump monitor* los nodos equipados con GPS y acelerómetro son utilizados para detectar baches en la carretera. Otros proyectos en este grupo son *Zebranet* [74] y *Manatee* [75] donde los nodos móviles monitorean el fenómeno, almacenan datos, viajan y cuando llegan a una ubicación específica (quizá algún lugar donde esté disponible un punto de acceso WiFi) transmiten sus datos hacia un servidor central. Todas estas aplicaciones mencionadas necesitan de costosa infraestructura (servicios de comunicación y servidores de datos) para transmitir la información y así trabajar apropiadamente.

En el *segundo grupo*, los nodos móviles utilizan infraestructura celular o WiFi para el envío y recepción de datos y adicionalmente pueden utilizar WiFi en modo ad hoc para solicitar a otros nodos que reenvíen sus datos a los servidores en caso de que dicha infraestructura celular o WiFi no esté disponible. Por ejemplo, en [141], los autores proponen una aplicación que enlaza las redes sociales con los teléfonos móviles implementando un protocolo en el que primero se comparten las identidades de los teléfonos móviles cercanos, usando *Bluetooth* o WiFi, y después cada teléfono consulta y descarga de las redes sociales datos relevantes acerca de las identidades recibidas con el objetivo de enriquecer la interacción humana. En [142], los autores proponen una red que permite a los usuarios de los teléfonos móviles enlazar una actividad como tomar una foto o muestrear los niveles de sonido cada hora con la ubicación física o área de interés. En esta aplicación, el usuario del teléfono difunde (*broadcast*) una solicitud de datos, los teléfonos cercanos reciben dicha solicitud, la retransmiten y si están en el área de interés monitorean el fenómeno y envían los datos a un servidor. El servidor recibe la información y enlaza o relaciona estos datos recibidos a la solicitud inicial; esto servirá para responder a algún otro teléfono que realice una solicitud similar. Otra aplicación en este grupo es CarTel [26]. En este trabajo los autores utilizan automóviles equipados con un equipo de cómputo acoplado con un conjunto de sensores, los cuales monitorean y procesan información acerca de las condiciones ambientales e infraestructura civil. Los datos procesados son enviados hacia un portal de información central a través de WiFi. En general, las soluciones en este segundo grupo aún utilizan costosa infraestructura, sin embargo también recurren a la comunicación utilizando WiFi en modo ad hoc entre nodos cercanos.

El *tercer grupo* incluye soluciones que no utilizan costosa infraestructura, en este grupo los nodos móviles comparten y solicitan información a otros nodos a través de WiFi en modo ad hoc, sin enviar o solicitar datos a algún servidor central. Por ejemplo, en [143] los autores proponen un sistema denominado *Traffic View*, donde los vehículos difunden su información a otros vehículos. Cada vehículo que recibe un paquete de datos de otro vehículo, actualiza dicho paquete con su propia información adquirida y lo reenvía a otros vehículos con el objetivo de compartir información acerca de las condiciones del tráfico, la planificación dinámica de rutas, mensajes de emergencia, etc. Esta solución está orientada a monitorear el tráfico y no necesita infraestructura para transmitir datos pero requiere que cada vehículo tenga GPS y una tarjeta de memoria. En [144], los autores proponen una aplicación que permite a la gente reservar taxis cercanos a su ubicación utilizando sus dispositivos móviles. En esta aplicación los dispositivos móviles de los usuarios y los que se encuentran en los taxis forman una red WiFi en modo ad hoc. Cuando un cliente necesita un taxi, comunica su solicitud directamente con los taxis que se encuentren dentro de su alcance de transmisión; los taxis que reciben esta solicitud la reenvían hasta que se encuentre un taxi disponible. En caso de que la petición del usuario no encuentre un taxi libre en modo ad hoc, la aplicación puede cambiar a un modo centralizado para solicitar el taxi a un servidor central. *Urbihoc* pertenece a este grupo, pero no requiere equipo especial, como equipos de cómputo, tarjetas de memoria adicionales, etc., y sólo trabaja en modo ad hoc. En general, para este tipo de soluciones la aplicación no debe ser muy sensible al retardo ya que, por la forma en la que operan, es posible que exista cierto retardo en la adquisición de la información. Una ventaja de este tipo de soluciones es que al no depender de infraestructura, éstas son menos costosas de implementar y pueden operar en lugares donde no exista infraestructura o cuando ésta haya sido afectada.

4.5 Urbihoc

Urbihoc es una solución para monitorear fenómenos en regiones grandes, por ejemplo una ciudad. La idea de *Urbihoc* consiste en obtener una vista completa de lo que sucede con el fenómeno de interés utilizando teléfonos móviles o vehículos como sensores [145]. Los datos recolectados son almacenados localmente y se comparten de manera oportuna con nodos móviles cercanos que estén ejecutando la aplicación. El intercambio de

información se lleva a cabo utilizando WiFi en modo ad hoc, por lo que con *Urbihoc* se logra un ahorro importante de recursos al no utilizar costosa infraestructura para realizar dicho intercambio.

La figura 4.3 ilustra el recorrido de dos nodos móviles que monitorean periódicamente el mismo fenómeno. A lo largo de dichas trayectorias estos nodos adquieren datos, los almacenan, y cuando se encuentran mutuamente dentro de sus alcances de cobertura, intercambian datos utilizando su WiFi en modo ad hoc. Al intercambiar los datos ambos nodos pueden saber cuáles son los valores medidos del fenómeno que les interesa de alguna región o zona que aún no han visitado, pero que quizá visiten pronto. De esta manera, *Urbihoc* permite que cada nodo tenga un conocimiento global del fenómeno a partir de los intercambios locales de datos con otros nodos. Debido a que el número de teléfonos móviles y de vehículos se ha incrementado rápidamente, es fácil imaginar que la información adquirida por cada nodo, gracias a los intercambios con otros sensores móviles, puede ser de gran utilidad.

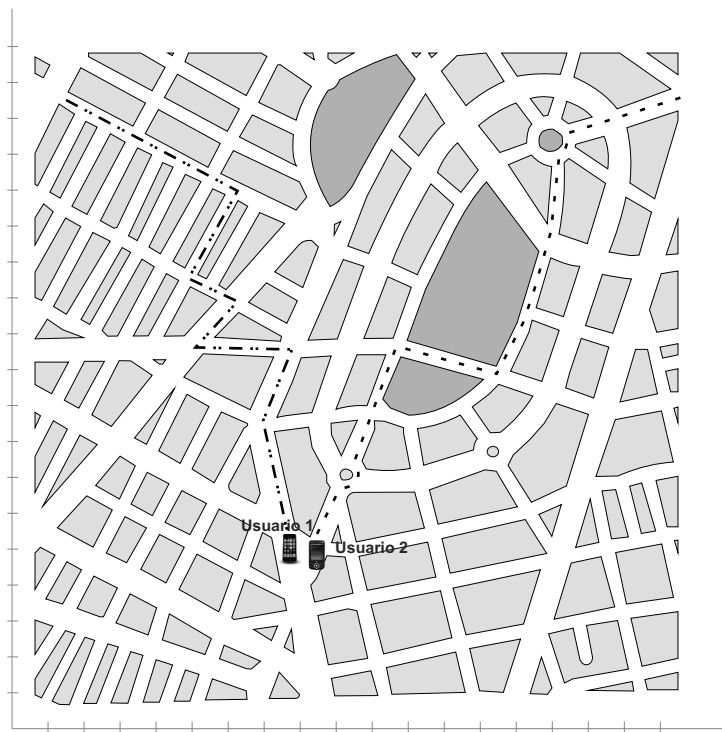


Figura 4.3: Trayectoria de dos nodos móviles

Se propone *Urbihoc* como una solución para satisfacer la necesidad de información acerca de cierto fenómeno de interés, sin que se inviertan muchos recursos. Como ya se mencionó, *Urbihoc* es adecuado para aquellos fenómenos cuyos valores son tolerantes al retardo y no se vuelven obsoletos rápidamente. La operación de *Urbihoc* consiste en tres principales etapas denominadas *clasificación*, *intercambio de conocimiento resumido* e *intercambio de solicitudes y conocimiento detallado*, mismas que son detalladas a continuación.

4.5.1 Clasificación

Después de que la aplicación *Urbihoc* es instalada en los nodos móviles, éstos monitorean periódicamente el fenómeno de interés y guardan cada uno de los valores observados en un registro z_i con la siguiente estructura $[k_i, (x_i, y_i), t_i]$, donde k_i es el valor de la variable monitoreada, (x_i, y_i) es la posición del nodo móvil en el momento de la lectura y t_i es la fecha y hora en la cual se realizó el monitoreo. Por la mayor disponibilidad que existe actualmente de los receptores del sistema GPS, se asume que los nodos móviles son capaces de determinar su ubicación, ya sea utilizando dichos receptores o algún otro método de localización [118, 146]).

En este capítulo, denominamos *conocimiento* a todos los datos adquiridos por cada nodo a lo largo de su trayectoria y éste puede ser representado de dos formas, forma *detallada* (Z) y forma *resumida* (S). La forma detallada contiene cada uno de los registros z_i y la resumida se construye a partir de promediar un número b de valores k que aparecen en dichos registros z_i . Por ejemplo, la figura 4.3 ilustra la ruta de dos nodos que tienen instalado *Urbihoc*, los puntos de ambas trayectorias indican las lecturas que cada nodo ha monitoreado. La tabla 4.1 muestra el conocimiento adquirido por el nodo 1 (U_1). En la primera columna se muestra cada registro z_i y al final de dicha columna se ilustra que el conocimiento, en modo detallado, es el conjunto de todos los registros z_i , $Z = \{z_i : i = 1, 2, \dots, n\}$. En la segunda columna se ejemplifica la construcción de cada elemento s_j que formará parte del conocimiento en modo resumido. La estructura de cada elemento s_j es $[(\sum_{i=1}^b k_i)/b, (x_1, y_1), (x_b, y_b), t_1, t_b]$, donde el primer elemento es el promedio de b valores monitoreados, para este caso $b = 5$, le siguen dos coordenadas (x_1, y_1) y (x_b, y_b) , que servirán para indicar el punto de inicio y fin en donde fueron monitoreados los valores; finalmente se encuentran t_1 y t_b que son el tiempo de inicio y fin de dicho monitoreo. Por ejemplo, el primer renglón en la segunda columna tiene

$s_1 = [(\sum_{i=1}^5 k_i)/5, (x_1, y_1), (x_5, y_5), t_1, t_5]$, este registro promedia 5 valores k_i contenidos en 5 registros z_i (de z_1 a z_5); (x_1, y_1) y (x_5, y_5) indican las coordenadas de inicio y fin, además se tiene t_1 y t_5 que se refieren al tiempo de inicio y fin de esos 5 registros z . Al final de la segunda columna se muestra que el conocimiento resumido contiene el conjunto de los registros s_j , $S = \{s_j : j = 1, 2, 3, \dots, m\}$.

4.5.2 Intercambio del conocimiento resumido (S)

Con frecuencia un nodo móvil se encontrará dentro de la región de cobertura WiFi de otros nodos móviles y de esta forma podrán crear un grupo temporal operando en modo ad hoc. Una vez que el grupo haya sido creado, cada nodo móvil que forme parte de dicho grupo prepara un paquete denominado p_a , el cual contiene el identificador del teléfono y su conocimiento resumido, con la siguiente estructura $p_a(id, S)$. Después de que todos los miembros del grupo han difundido su paquete p_a , cada teléfono sabrá cuál es el comportamiento promedio del fenómeno de interés y un aproximado del área en la cual se monitoreó dicha información. El área es aproximada porque, como ya se mencionó, en el conocimiento resumido sólo se tienen las coordenadas de inicio y fin. Por ejemplo, en la Figura 4.4 hay tres nodos móviles agrupados (U_1, U_2, U_3) y cada uno difunde (*broadcast*) un paquete p_a .

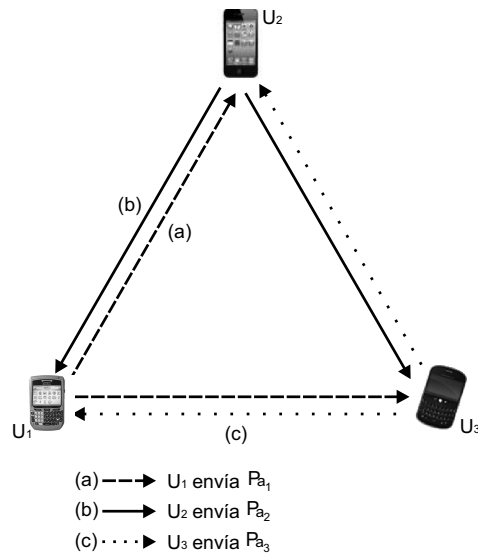


Figura 4.4: Cada nodo difunde un paquete p_a que contiene su ID y su conocimiento resumido S .

Formas de conocimiento	
Detallado (Z)	Resumido (S)
$z_1 = [k_1, (x_1, y_1), t_1]$	
...	
$z_5 = [k_5, (x_5, y_5), t_5]$	$s_1 = [(\sum_{i=1}^5 k_i)/5, (x_1, y_1), (x_5, y_5), t_1, t_5]$
$z_6 = [k_6, (x_6, y_6), t_6]$	
...	
$z_{10} = [k_{10}, (x_{10}, y_{10}), t_{10}]$	$s_2 = [(\sum_{i=6}^{10} k_i)/5, (x_6, y_6), (x_{10}, y_{10}), t_6, t_{10}]$
...	...
$z_{n-b} = [k_{n-b}, (x_{n-b}, y_{n-b}), t_{n-b}]$	
...	
$z_n = [k_n, (x_n, y_n), t_n]$	$s_m = [(\sum_{i=n-b}^n k_i)/b, (x_{n-b}, y_{n-b}), (x_n, y_n), t_{n-b}, t_n]$
$Z = \{z_i : i = 1, 2, 3, \dots, n\}$	$S = \{s_j : j = 1, 2, 3, \dots, m\}$

Tabla 4.1: Conocimiento adquirido por el nodo U_1

Continuando con el ejemplo ilustrado en la figura 4.4, después de que los nodos del grupo han difundido su conocimiento resumido, cada nodo tendrá como conocimiento lo que se muestra en la tabla 4.2. En el caso del nodo U_1 , éste posee su propio conocimiento Z_1 y además ahora tiene el conocimiento resumido S_2 y S_3 , que recibió de U_2 y U_3 , respectivamente.

4.5.3 Intercambio de solicitudes y conocimiento detallado

Como ya se mencionó, después de la difusión de paquetes p_a cada nodo móvil puede determinar cuál es el comportamiento promedio del fenómeno de interés en ciertas áreas. Si esta información es suficiente para los usuarios de los nodos móviles, el intercambio

Nodos	Datos
U_1	Z_1, S_2, S_3
U_2	Z_2, S_1, S_3
U_3	Z_3, S_1, S_2

Tabla 4.2: Conocimiento adquirido por los nodos móviles U_1 , U_2 y U_3

de datos termina; de lo contrario, si alguno de los nodos está interesado en conocer con mayor detalle la información recibida, entonces el nodo difunde un paquete de solicitud denominado p_b . Hemos definido cuatro tipos de solicitudes y por ende cuatro tipos de paquetes p_b . El primer tipo se utiliza para solicitar a todos los nodos del grupo toda la información que desconozca el nodo que está difundiendo dicha solicitud $p_{b_1}(id, ALL)$; el segundo tipo se utiliza para solicitar toda la información pero sólo de ciertos nodos, $p_{b_2}(id, ALL-id_i)$; el tercer tipo se utiliza para solicitar el detalle de determinada información de ciertos nodos, $p_{b_3}(id, S_i-id_i, S_j-id_j)$; el cuarto tipo permite mezclar y así poder solicitar toda la información de un nodo y sólo parte de otro, $p_{b_4}(id, ALL-id, S_i-id_i)$. Cuando un nodo recibe alguno de los paquetes de solicitud antes mencionados, prepara un paquete p_c con la información detallada solicitada y lo envía.

4.5.4 Cantidad de información almacenada

Debido a que el espacio de almacenamiento en los nodos móviles está limitado, es adecuado almacenar sólo un periodo de información. En *Urbihoc* dicho periodo es controlado mediante el parámetro α , el cual establece el número máximo de registros r que pueden ser almacenados. El proceso de almacenamiento consiste en lo siguiente:

- Cada nodo móvil mantendrá un número consecutivo i de sus registros z_i .
- Se utilizará un arreglo circular [147, 148, 149] para almacenar cada registro z_i .
- El resultado de la operación $i \bmod \alpha$ indicará en qué posición del arreglo debe ser almacenado el nuevo registro z .

4.5.5 Estimación de datos

Como ya se mencionó, resulta costoso lograr una cobertura completa en la que al menos un dispositivo monitoree cada punto o zona de algún fenómeno de interés en un área grande. Sin embargo, aunque el despliegue de sensores no logre una cobertura completa, es posible estimar los valores de los puntos o zonas intermedias ubicadas entre sensores cercanos utilizando algún método de interpolación [150, 151, 152].

La interpolación [153] es el proceso de utilizar datos conocidos de ciertas zonas para estimar valores desconocidos de otras zonas. Existen varios métodos de interpolación [153], por ejemplo la interpolación que utiliza el valor del *vecino más cercano* para estimar algún valor desconocido. Otro tipo es la *interpolación bilineal* en la cual se utilizan los valores de cuatro vecinos cercanos para estimar el valor desconocido usando la ecuación 4.1

$$v(x, y) = ax + by + cxy + d \quad (4.1)$$

Un tercer tipo es la *interpolación bicúbica*, la cual involucra valores de dieciséis vecinos cercanos para realizar dicha estimación, mediante el uso de la ecuación 4.2

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \quad (4.2)$$

En *Urbihoc* se puede utilizar cualquiera de estas tres formas de interpolación, por ejemplo en la figura 4.5(a) se ilustra nuevamente la información adquirida por el nodo identificado con el número 50 y en la figura 4.5(b) se ilustra el conocimiento detallado adquirido gracias a los intercambios realizados con otros nodos. En la figura 4.5(c) se muestra el resultado de aplicar la interpolación bicúbica dada, la ecuación 4.2, para estimar los valores de los datos faltantes a partir del conocimiento detallado.

4.6 Evaluación del desempeño

En esta sección se presenta una evaluación de *Urbihoc* y se discuten algunos problemas de rendimiento asociados con la movilidad del usuario, la obsolescencia de la información y la optimización de la energía.

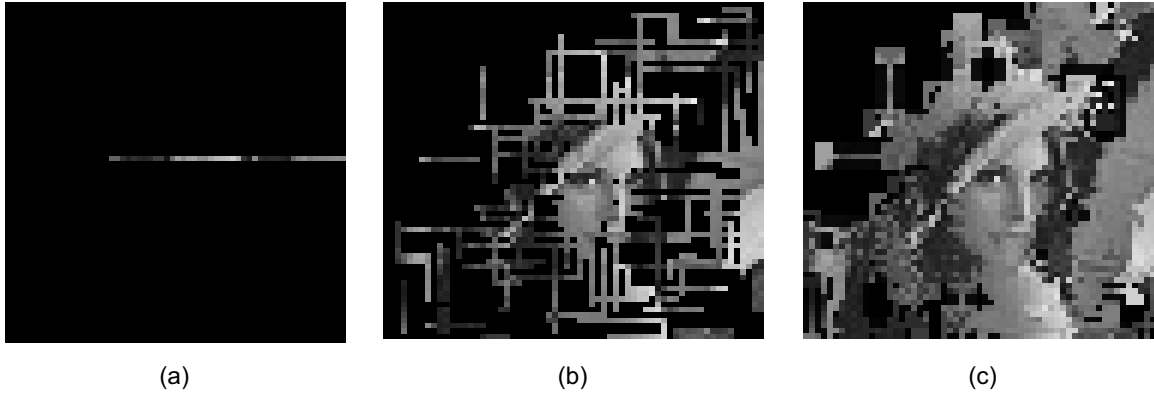


Figura 4.5: (a) Conocimiento adquirido por el nodo identificado con el número 50. (b) Conocimiento detallado ($b = 1$) adquirido por el nodo 50. (c) Conocimiento interpolado del nodo 50 a partir del conocimiento detallado.

4.6.1 Simulaciones

Para evaluar el desempeño de *Urbihoc*, se utilizó el simulador NS 2.29 [154]. Para representar el fenómeno se utilizó la imagen que se muestra en la figura 4.6(a) y sobre ésta se colocaron aleatoriamente 100 nodos móviles. También aleatoriamente, cada nodo seleccionó una serie de destinos y se desplazó hacia cada uno de éstos siguiendo trayectorias predefinidas y con una velocidad promedio de 3 km/h; la cual corresponde a usuarios caminando. El modelo que se utilizó está basado en el modelo de movilidad en caminos (*Pathway Mobility Model*) [155, 156, 157, 158, 159], el cual es un modelo con restricción geográfica en donde los nodos se mueven siguiendo los caminos y direcciones establecidas en el mapa que se utilice. En cada movimiento los nodos monitorearon el fenómeno y por cada lectura almacenaron un nuevo registro z_i . Cada 10 segundos, todos los nodos se asociaron de ser posible en grupos temporales con otros nodos localizados dentro de su alcance de transmisión e intercambiaron información. Se realizaron dos tipos de simulaciones, en el primer tipo los nodos sólo intercambiaron el conocimiento resumido S con una $b = 5$. En el segundo tipo intercambiaron sólo el conocimiento detallado (Z) ($b = 1$). Para realizar los intercambios se utilizó el estándar IEEE 802.11g (sin emplear periodos de inactividad en los nodos). Se obtuvieron diversos resultados y se muestran, como ejemplo, los resultados relacionados al nodo identificado con el número 50. La figura 4.6(b) ilustra los datos que dicho nodo adquirió por sí mismo, la figura 4.6(c) muestra el conocimiento resumido adquirido por este nodo a partir de los intercambios y la figura

4.6(d) muestra el resultado de interpolar dicho conocimiento resumido. La figura 4.6(e) muestra el conocimiento detallado adquirido también por los intercambios de datos con los nodos cercanos y la figura 4.6(f) muestra el resultado de interpolar el conocimiento detallado.

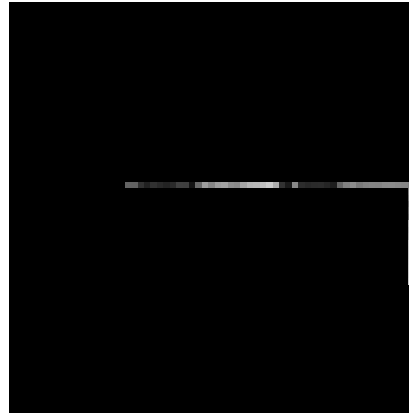
Al comparar las figuras 4.6(b, c y e) se aprecia la importante utilidad de compartir información. En la figura 4.6(b) se ilustra que el nodo 50 adquirió por sí mismo aproximadamente el 1.2 % del total de la información, mientras que en las figuras 4.6(c y e) se aprecia que el nodo aprendió aproximadamente el 45 % del total de los datos después de realizar 5 intercambios (cada 10 movimientos se realizó un intercambio). Las figuras 4.6(d y f) muestran el resultado de interpolar el conocimiento resumido y detallado de las figuras 4.6(c y e), respectivamente. Dicho proceso de estimación del conocimiento permitió adquirir en promedio un 30 % más de información, lo cual puede apreciarse en las figuras 4.6(d y f).

Para calcular la similitud entre la información obtenida respecto a la información original se utilizó una correlación cruzada [153]. La figura 4.7 ilustra el porcentaje de similitud obtenido, en dicha figura se aprecia que el mayor porcentaje ocurre cuando se interpola el conocimiento detallado pues el parecido es un poco más del 50 % y la información que tiene menor similitud es la que el nodo adquirió por sí mismo, pues se logra sólo un 5 % de parecido.

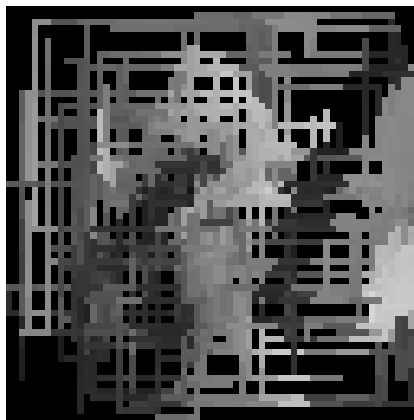
La figura 4.8 muestra el número de bytes de conocimiento resumido, utilizando una $b = 5$, que recibió y transmitió el nodo 50, así como el acumulado del conocimiento total adquirido. Se observa que en los primeros 6 intercambios el nodo 50 recibe en promedio 15,000 bytes, conforme adquiere más conocimiento la información que desconoce es menor y por ello disminuye el número de bytes que recibe. También se aprecia que el número de bytes que transmite a veces es alto y otra vez es bajo, esto se debe a que conforme el nodo 50 se desplaza a otras áreas es probable que se encuentre con nodos que desconocen la información que éste posee y por ello se incrementa el número de bytes transmitidos, siendo el más alto 90,000 bytes y 1000 bytes el menor.



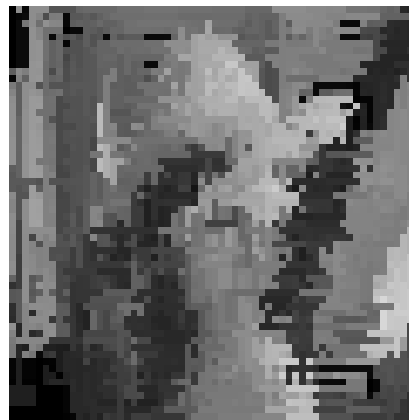
(a)



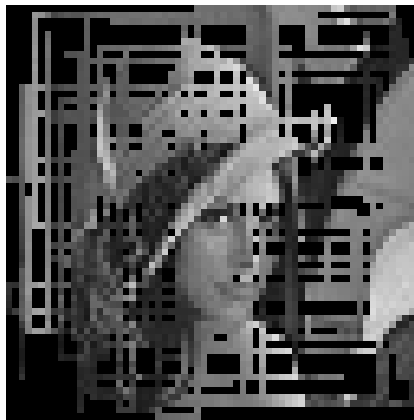
(b)



(c)



(d)



(e)



(f)

Figura 4.6: (a) Imagen utilizada para representar el fenómeno, sobre ésta se colocaron aleatoriamente 100 nodos móviles. (b) Información adquirida por el nodo identificado con el número 50. (c) Conocimiento resumido (S) adquirido por el nodo 50 utilizando $b = 5$. (d) Interpolación del conocimiento resumido. (e) Conocimiento detallado (Z) ($b = 1$) adquirido por el nodo 50. (f) Interpolación del conocimiento detallado.

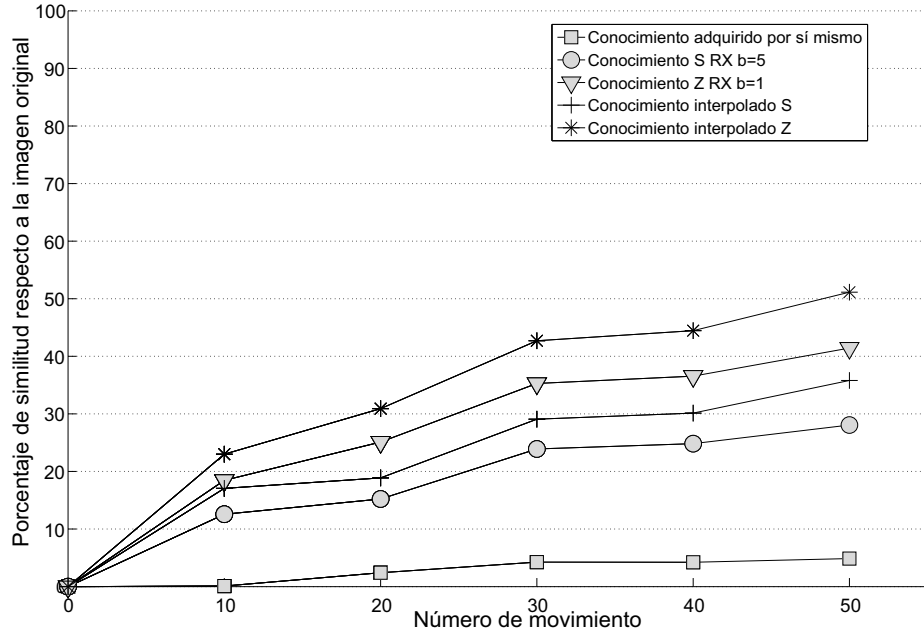


Figura 4.7: Porcentaje de similitud entre la información original y la información obtenida.

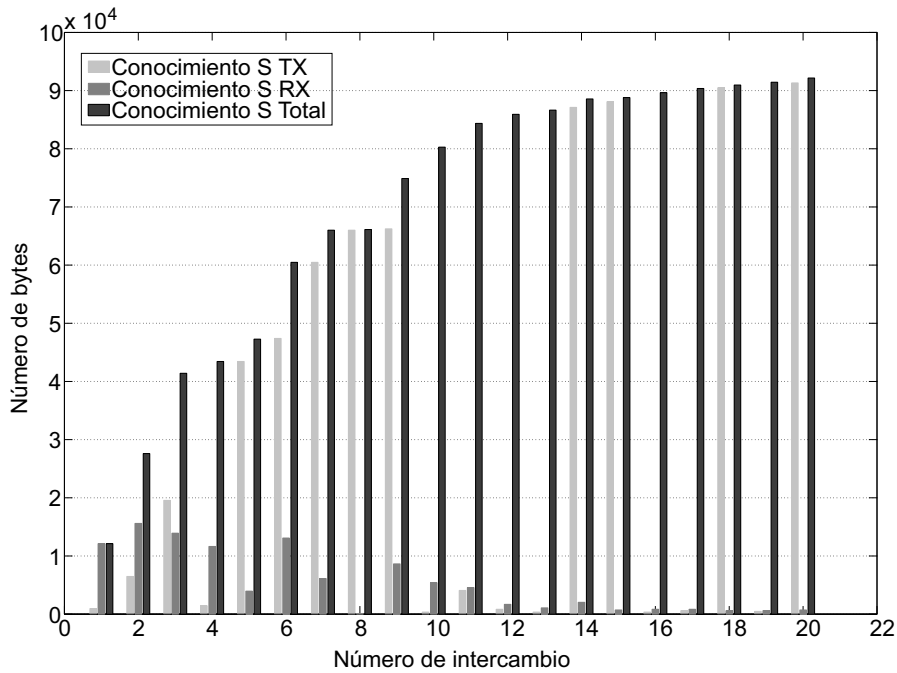


Figura 4.8: Número de bytes transmitidos, recibidos y total de conocimiento resumido (S) ($b = 5$) por el nodo 50.

En la figura 4.9 se ilustra el número de bytes de conocimiento detallado que envía y recibe el mismo nodo 50, así como el conocimiento total acumulado. En esta figura se observa que en los primeros 6 intercambios el nodo recibe en promedio 40,000 bytes. El número más alto de bytes transmitidos es de 250,000 y esto ocurre cuando el nodo adquiere más conocimiento. Con las figuras 4.8 y 4.9 se observa que si el usuario desea una visión general de lo que sucede con el fenómeno de interés, es buena opción intercambiar únicamente el conocimiento resumido, pues si comparamos las figuras 4.6(c y e), se puede apreciar que la figura 4.6(c) no tiene el detalle que tiene la figura 4.6(e), pero es suficiente para tener una idea general de cómo se está comportando el fenómeno de interés. A mayor detalle, mayor es el número de transmisiones y recepciones y en consecuencia mayor es el gasto de energía.

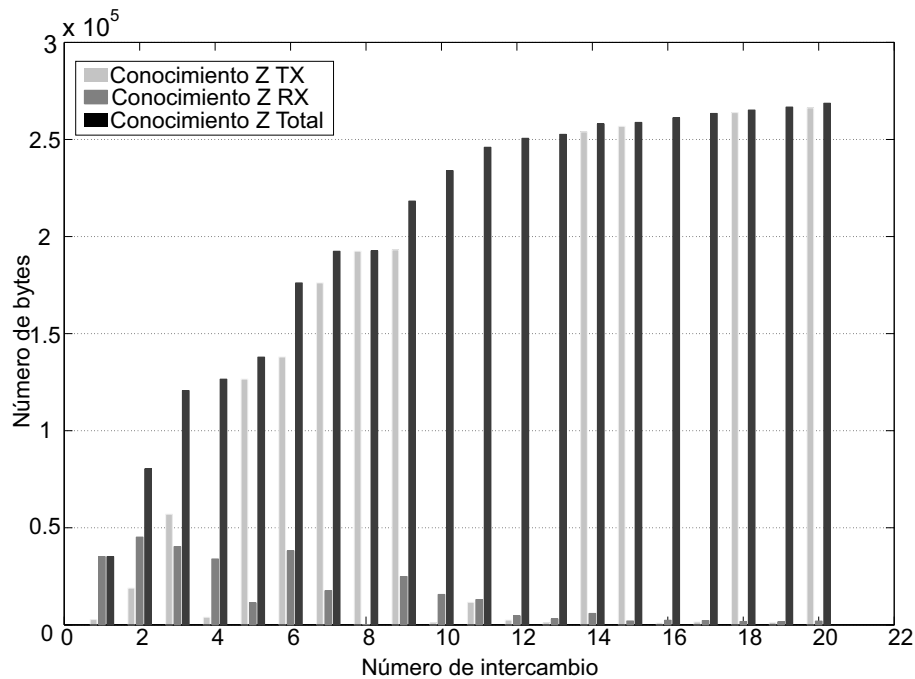
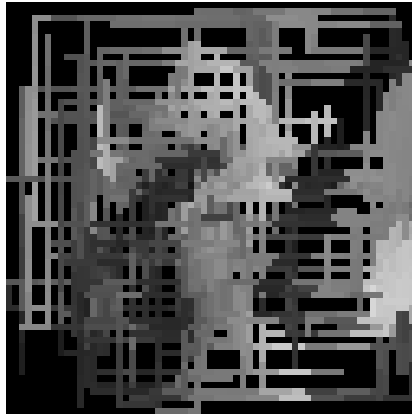
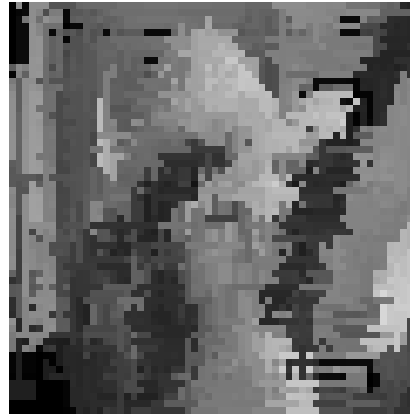


Figura 4.9: Número de bytes transmitidos, recibidos y total de conocimiento detallado (Z) por el nodo 50.

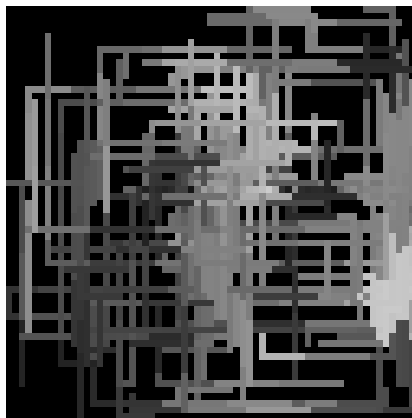
Si el valor de b se incrementa, disminuye el detalle con que se construye el conocimiento resumido y por el contrario, si el valor de b disminuye entonces se incrementa el detalle de dicho conocimiento. Por ejemplo, en las figuras 4.10(a, c y e) se ilustra el conocimiento resumido adquirido por el nodo 50 utilizando una b con valores de 5, 10 y 15, respectivamente, y también se muestra el resultado de interpolar dicho conocimiento en las figuras 4.10(b, d y f).



(a)



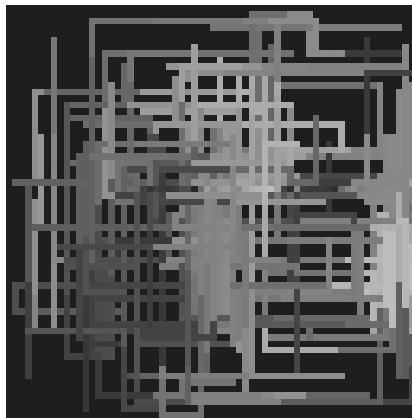
(b)



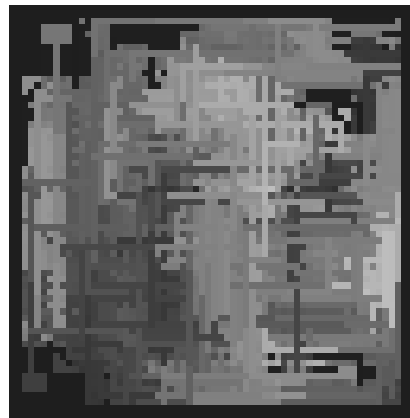
(c)



(d)



(e)



(f)

Figura 4.10: (a) Conocimiento resumido con $b = 5$ y su interpolación (b). (c) Conocimiento resumido con $b = 10$ y su interpolación (d). (e) Conocimiento resumido con $b = 15$ y su interpolación (f).

4.6.2 Implementación

Urbihoc se implementó utilizando *Android* 4.0 API 14 [148, 149, 160], por cuestiones prácticas, como nodos móviles se utilizaron tres modelos diferentes *smartphones*: *LG Optimus L7-9708*[161], *Samsung Galaxy SIII-GT-I9300*[162] y *Motorola Razr*[163] evaluando así la viabilidad de la propuesta en escenarios con cierta heterogeneidad en cuanto a los dispositivos móviles utilizados.

Dado que todos los equipos utilizados contaban con sensores de luminosidad, se optó por monitorear como fenómeno de interés, los niveles de intensidad de luz ambiental (flujo luminoso) generada por el alumbrado público en una determinada zona de la Ciudad de México. Para esto se realizaron monitoreos nocturnos, para identificar zonas de obscuridad o zonas en donde las lámparas del alumbrado público son muy viejas y aunque encienden, la intensidad de luz que proporcionan es poca y, por ende, deben ser sustituidas. Se instaló la aplicación de *Urbihoc* en 5 nodos los cuales realizaron distintas trayectorias. Por ejemplo, la figura 4.11 muestra la trayectoria del usuario con el nodo móvil identificado con el número 1 y la figura 4.12 muestra gráficamente las lecturas tomadas por dicho teléfono. Las lecturas o mediciones tomadas (f) por los nodos se clasificaron con base en qué tan iluminado observamos el área monitoreada, la tabla 4.3 muestra dicha clasificación.

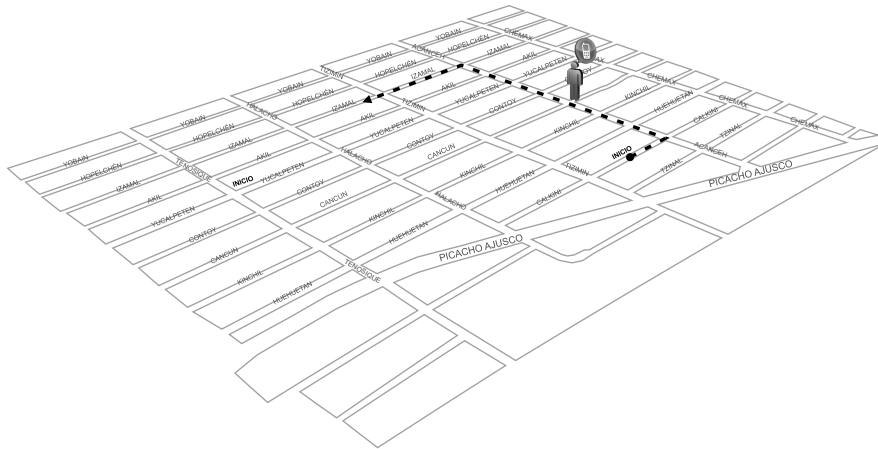


Figura 4.11: Trayectoria del usuario con el nodo móvil identificado con el número 1.

Lectura (f) en luxes	Flujo luminoso
$1 \leq f \leq 20$	Poca
$20 < f \leq 40$	Mediana
$40 < f \leq 80$	Buena
$f > 80$	Muy buena

Tabla 4.3: Clasificación del nivel de iluminación

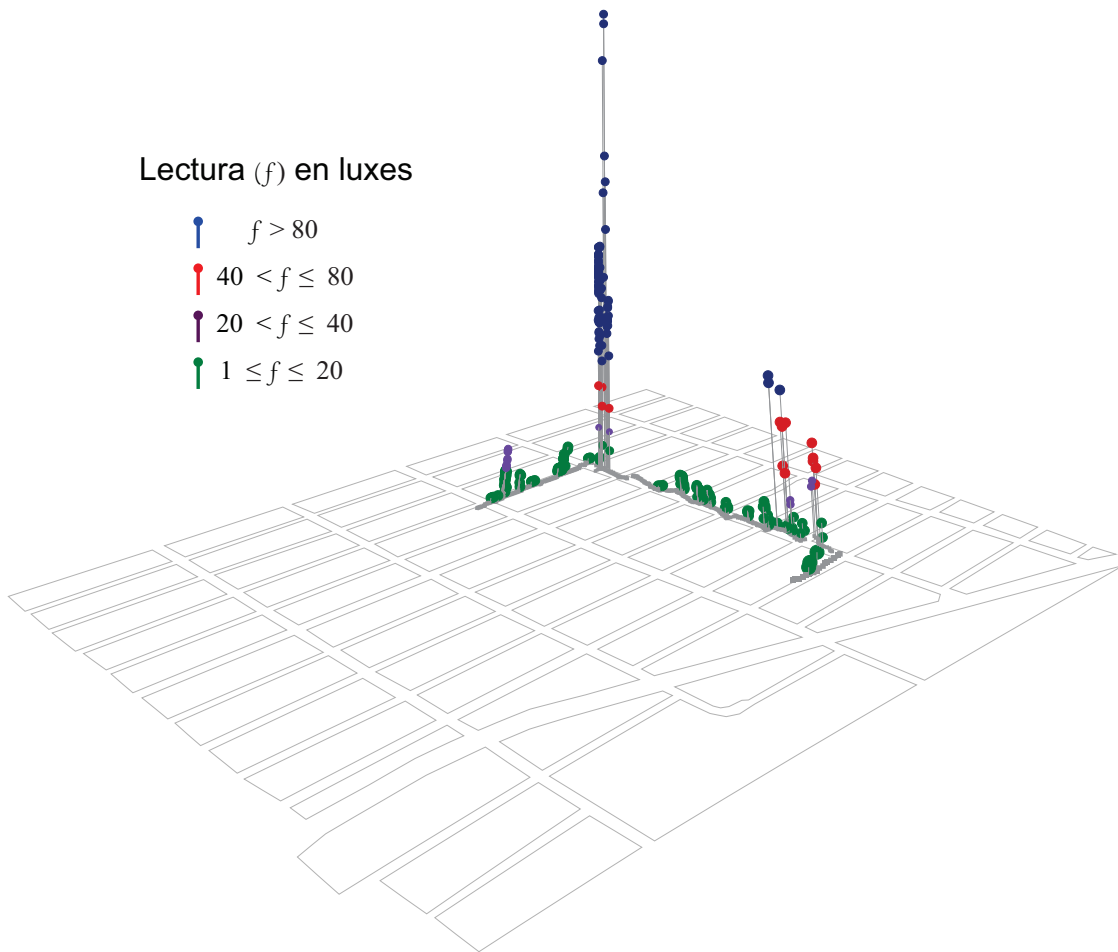


Figura 4.12: Conocimiento adquirido individualmente por el nodo 1 durante toda su trayectoria.

Durante su trayectoria el nodo 1 se encontró con el usuario del nodo 2 (ver figura 4.13) e intercambiaron su conocimiento resumido y detallado. Al finalizar el intercambio, el conocimiento del nodo 1 se incrementó debido a los datos que recibió del nodo 2 y se ilustra en la figura 4.14. Al final del recorrido el nodo 1 adquirió el conocimiento ilustrado en la figura 4.15.

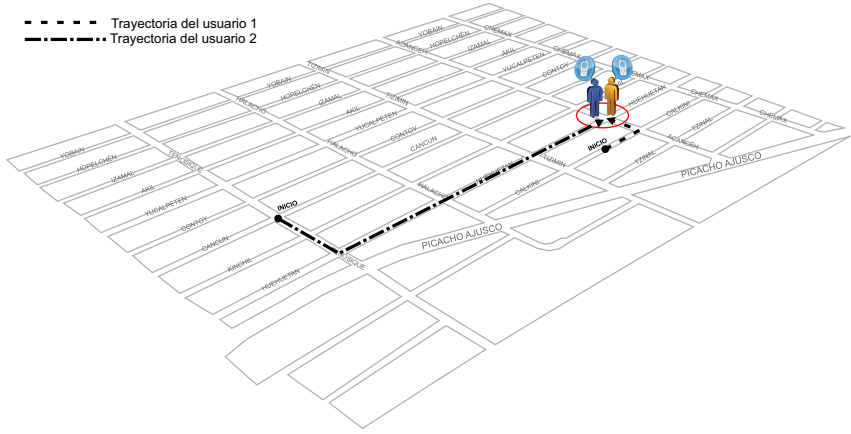


Figura 4.13: Parte de las trayectorias de los usuarios 1 y 2.

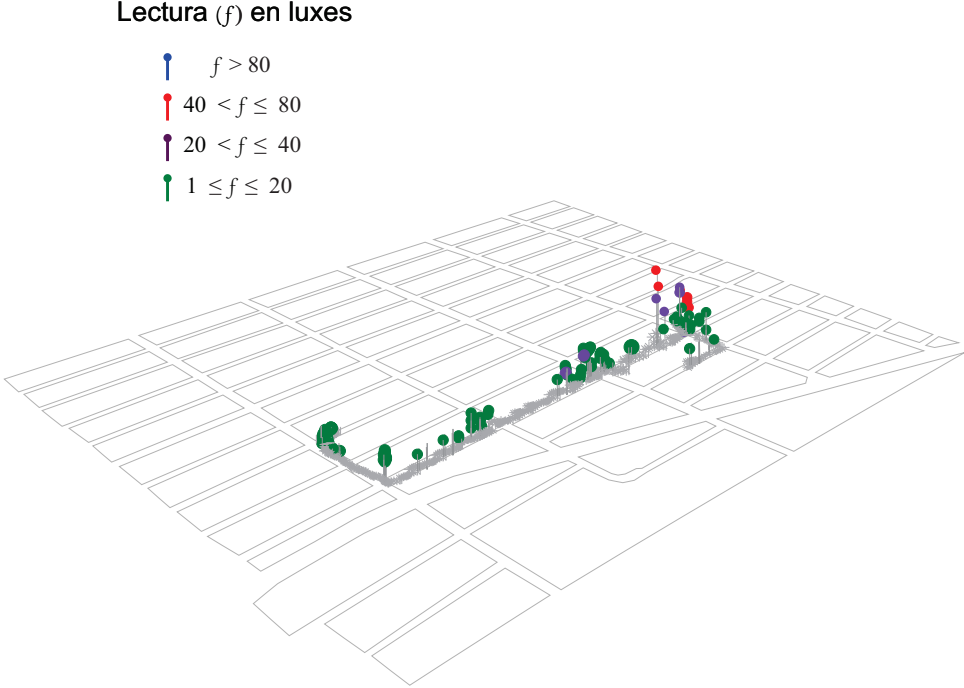


Figura 4.14: Conocimiento adquirido por el nodo 1 después de su primer intercambio.

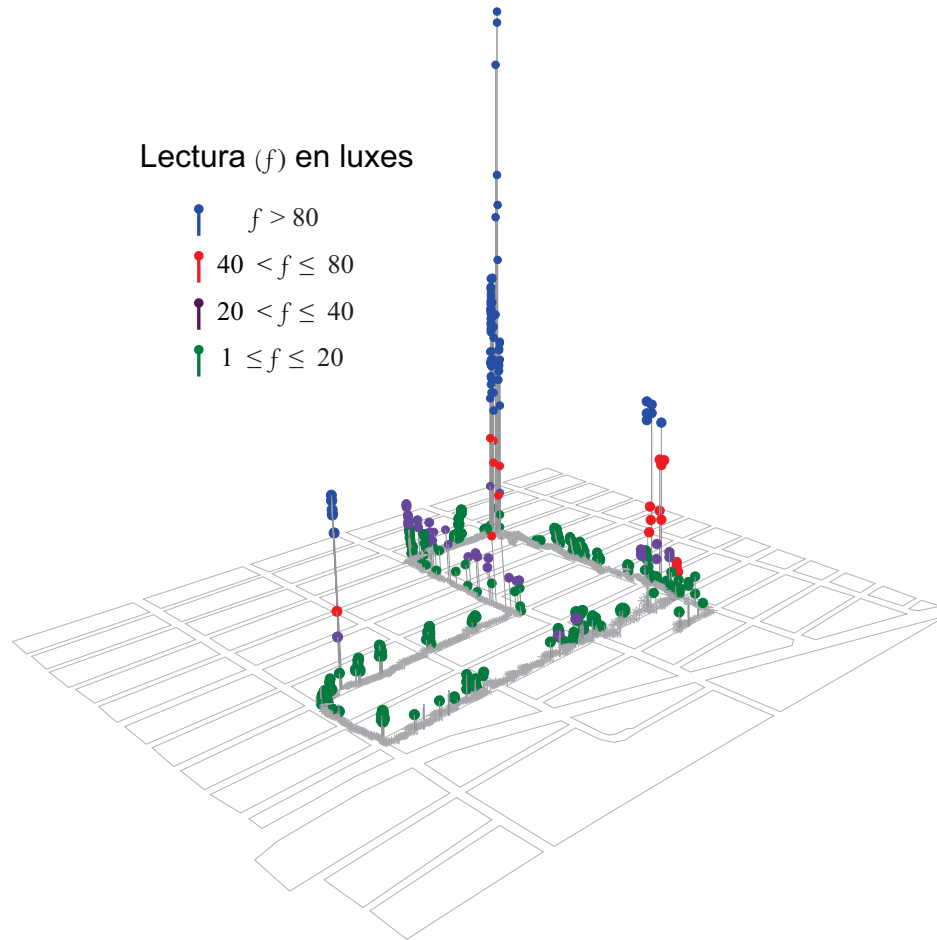


Figura 4.15: Total de conocimiento adquirido por el nodo 1.

4.7 Conclusiones del capítulo

En este trabajo se propone *Urbihoc*, un nuevo método para la adquisición de datos en las redes inalámbricas de sensores, el cual utiliza teléfonos móviles como sensores y aprovecha la movilidad de los usuarios de dichos teléfonos para adquirir una vista completa de lo que sucede con el fenómeno de interés sin el uso de infraestructura costosa. A diferencia de otras soluciones, en *Urbihoc* los teléfonos no envían datos hacia un servidor central y no utilizan la infraestructura celular o servicios WiFi para compartir su información. En *Urbihoc* los teléfonos comparten sus datos adquiridos exclusivamente en modo ad hoc utilizando comunicación directa para realizar intercambios locales y así obtener un cono-

cimiento global acerca del fenómeno monitoreado. Debido a la movilidad de los usuarios, en el intercambio inicial cada teléfono envía información en modo resumido (pocos *bytes*) para que dicho intercambio se realice rápidamente. Los datos resumidos pueden ser interpolados por el teléfono móvil receptor para así lograr un mayor conocimiento acerca del fenómeno. Además, existe una segunda posibilidad que consiste en intercambiar el conocimiento detallado cuando así lo soliciten los teléfonos. En este capítulo se demuestra que este comportamiento es la esencia de una técnica eficiente en cuestión de que no se utiliza infraestructura costosa y que permite obtener una vista completa de lo que sucede con el fenómeno de interés.

Urbihoc fue implementado en el simulador NS2.29 y se llevaron a cabo varios experimentos para probar su factibilidad. Los resultados indican que para tener una visión general del fenómeno es suficiente con intercambiar la información resumida e interpolarla al recibirla. *Urbihoc* se ve afectado por la movilidad y la rapidez con la que caduquen los datos del fenómeno en cuestión, por ello es recomendable para aquellos fenómenos cuyos valores no se vuelvan obsoletos rápidamente como son la medición de la temperatura, de los niveles de contaminación y la detección de la presencia de polén en regiones urbanas.

Capítulo 5

Conclusiones y trabajo futuro

Una de las tareas más importantes en las redes inalámbricas de sensores es la recolección de datos. Como resultado de este trabajo de investigación, se proponen algunas estrategias que permiten eficientar dicho proceso de recolección a través del ahorro de ciertos recursos tales como la energía, infraestructura, ancho de banda entre otros.

El ahorro de energía es un aspecto crítico en el diseño y desempeño de las redes inalámbricas de sensores. Obtener información completa de lo que sucede con el fenómeno de interés en toda un área, es una actividad que normalmente involucra un gasto de energía considerable, debido a que todos los nodos deben enviar sus lecturas adquiridas hacia el nodo *sink*, y los nodos intermedios deben reenviar los paquetes que reciben de otros nodos. Esto no ocurre con nuestra propuesta, en la cual la solicitud de los datos se ajusta a los valores de cada vecindario permitiendo así explotar la correlación espacial a nivel de sensor. Con esta forma de solicitud sólo los nodos cuyos datos estén fuera del rango solicitado serán los únicos que transmitan su información, mientras que los demás permanecerán en silencio, logrando con esto un importante ahorro de energía.

Esta disertación presenta sólidos argumentos para demostrar que en caso de que el fenómeno que se requiere monitorear tenga correlación espacial, se utilicen sensores estáticos y se necesite periódicamente una vista completa de lo que sucede con dicho fenómeno, no es necesario que todos los sensores transmitan sus datos. En el capítulo 3 se muestra a través de simulaciones que *Disagree* es capaz de ahorrar energía en los sensores aun cuando la correlación espacial en el fenómeno es baja. Los resultados expe-

rimentales muestran que *Disagree* es más eficiente, en cuestión de ahorro de energía, en comparación con otros métodos reportados en la literatura. En *Disagree*, el problema de que a los sensores cercanos al nodo *sink* se les agota la batería rápidamente por todo el trabajo de reenvío que realizan, se resuelve mediante la rotación periódica de los nodos, esto aunado a la existencia de nodos silenciosos, permite extender el tiempo de vida de la red. Creemos que la forma de operar de *Disagree* puede ser incluida en otros protocolos, como LEACH [30], TEEN [59] y APTEEN [105], para lograr un mayor ahorro de energía. Debido a la naturaleza de las redes de sensores estáticos, *Disagree*, al igual que otras soluciones de su categoría, no se puede escalar fácilmente para ser utilizado en el monitoreo de fenómenos en áreas muy extensas, como por ejemplo una ciudad, ya que se requeriría de muchos sensores y las tareas de mantenimiento y supervisión de éstos se volvería poco viable. Por ello decidimos introducir sensores móviles para lograr una mayor cobertura sin necesidad de incrementar en demasía el número de dispositivos y no complicar su mantenimiento.

El uso de sensores móviles mejora la cobertura permitiendo así el monitoreo de diversos fenómenos en áreas grandes. Para obtener una vista completa de lo que sucede con el fenómeno de interés, los sensores móviles pueden transmitir sus datos adquiridos hacia una estación central utilizando la red celular o los servicios WiFi disponibles sin embargo, la implementación y el acceso de este tipo de infraestructura es muy costoso. En el capítulo 4 se introdujo *Urbihoc*, un método flexible, económico y escalable que permite a cada nodo móvil obtener una vista completa de lo que sucede con algún fenómeno en un área extensa, a partir de intercambios locales con nodos vecinos sin necesidad de utilizar infraestructura costosa. En *Urbihoc* los nodos móviles pueden ser vehículos particulares, unidades de transporte público, teléfonos celulares y algunos otros dispositivos que incluyan sensores y puedan monitorear el fenómeno en cuestión. La movilidad diaria de los nodos permite el encuentro con otros nodos llevando a cabo el intercambio de información, y de esta forma obtener un conocimiento global acerca del fenómeno monitoreado. Para los experimentos y por cuestiones prácticas, decidimos utilizar teléfonos celulares como nodos móviles [164, 27]. A través de simulaciones se demostró que realizar intercambios locales entre nodos, permite que el usuario conozca cuál es el comportamiento de la variable monitoreada en determinado lugar sin necesidad de haber estado ahí. *Urbihoc* presenta algunas limitantes tales como la impredecibilidad de las rutas, lo

cual ocasiona que no se pueda garantizar una cobertura completa del escenario a pesar de que haya una buena densidad de nodos en la red. Adicionalmente, debido a que no se utiliza costosa infraestructura, los nodos deben almacenar localmente sus lecturas y quizá cuando éstas son compartidas con los nodos vecinos ya no son tan actuales. Por ello es que *Urbihoc* es adecuado para el monitoreo de fenómenos cuya información no caduque rápidamente.

Creemos que es posible construir una solución híbrida, en la que existan nodos estáticos y nodos móviles de tal manera que los nodos móviles, ejecutando *Urbihoc*, podrían entregar la información recolectada a un nodo estático, el cual podría reportar datos hacia una estación central utilizando *Disagree*. Debido a que los nodos móviles pueden ser unidades de transporte público, podrían aprovecharse las rutas que éstos siguen diariamente para difundir y recolectar información a lo largo de alguna ciudad, realizando esta adquisición en breve tiempo.

Como se mencionó, el tiempo de contacto entre nodos móviles es impredecible pues depende de la velocidad a la que éstos se desplazan, por ello se requiere trabajar en diversas estrategias para lograr intercambios exitosos de datos en el menor tiempo posible y así incrementar el número de situaciones en las que los nodos móviles tipo automóvil pueden intercambiar datos con los nodos móviles tipo teléfono. También es necesario trabajar en estrategias para motivar a los usuarios a compartir sus datos, asegurando su privacidad ya que por ejemplo, ningún usuario estará de acuerdo en compartir sus datos si otro usuario puede deducir la ruta que siguió, porque quizá esa sea su ruta diaria y dicha información podría utilizarse con otros fines. Es posible integrar y fortalecer la propuesta de *Urbihoc* con las estrategias antes mencionadas.

Anexo A

Lista de publicaciones

Artículos de revista

M. Montes-de-Oca, J. Gomez, M. Lopez-Guerrero, "DISAGREE: Disagreement-based querying in wireless sensor networks", *Telecommunications Systems*, Springer, ISSN: 1018-4864, doi:10.1007/s11235-013-9852-5, pp.1-18, 2013.

M. Gonzalez, J. Gomez, V. Rangel, M. Lopez-Guerrero, M. Montes-de-Oca, "GUIDE-Gradient: A Guiding Algorithm for MobileNodes in WLAN and Ad hoc Networks", *Wireless Personal Communications*, Springer ISSN: 0929-6212, Vol. 57, No. 4, pp. 629-653, 2011.

Artículos de conferencia

C. You, M. Montes-de-Oca, T. Bao, N. Lane, H. Lu, G. Cardone, L. Torresani, A. T. Campbell, "CarSafe Demo: Supporting Driver Safety Using Dual-cameras on Smartphones", *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 547-547, 2012.

C. You, N. Lane, F. Chen, R. Wang, Z. Chen, T. Bao, M. Montes-de-Oca, Y. Cheng, M. Lin, L. Torresani, A. Campbell, "CarSafe App: Alerting Drowsy and Distracted Drivers using Dual Cameras on Smartphones", *Proceedings of the 11th annual international conference on Mobile systems, applications, and services*, pp. 461-462, 2013.

Bibliografía

- [1] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2007.
- [2] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective*. John Wiley & Sons, 2009.
- [3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [4] G. Amitabha and K. Sajal, *Coverage and Connectivity Issues in Wireless Sensor Networks*, pp. 221–256. John Wiley & Sons, 2006.
- [5] L. Benyuan and D. Towsley, “A study on the coverage of large-scale sensor networks,” in *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, (Florida, EUA), pp. 475–483, 2004.
- [6] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, “Coverage problems in wireless ad-hoc sensor networks,” in *Proceedings of the 20th Annual Conference of IEEE Computer and Communications Societies*, (Anchorage, AK), pp. 1380–1387, 2001.
- [7] R. Shorey, A. Ananda, M. C. Chan, and W. T. Ooi, *Mobile, Wireless, and Sensor Networks: Technology, Applications, and Future Directions*. John Wiley & Sons, 2006.
- [8] D. Gage, “Command Control for Many-robot Systems,” in *Proceedings of the 1992 Annual Unmanned Vehicle Systems Technical Symposium*, (Huntsville, AL, EUA), pp. 28–42, 1992.

- [9] E. Onur, C. Ersoy, and H. Delic, “How many sensors for an acceptable breach detection probability?,” *Computer Communications*, vol. 29, no. 2, pp. 173–182, 2006.
- [10] G. Fan, R. Wang, H. Huang, L. Sun, and C. Sha, “Coverage-Guaranteed Sensor Node Deployment Strategies for Wireless Sensor Networks,” *Sensors*, vol. 10, no. 3, pp. 2064–2087, 2010.
- [11] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, “Mobility Improves Coverage of Sensor Networks,” in *Proceedings of the 6th ACM International symposium on Mobile ad hoc networking and computing*, (Illinois, EUA), pp. 300–308, 2005.
- [12] B. Ren, W. Jiao, B. Wang, D. Xie, and J. Ma, “Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing,” in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, (Ontario, Canada), pp. 113–120, 2007.
- [13] B. Wang, K. Chua, V. Srinivasan, and W. Wang, “Sensor Density for Complete Information Coverage in Wireless Sensor Networks,” in *Wireless Sensor Networks*, vol. 3868 of *Lecture Notes in Computer Science*, pp. 69–82, Springer Berlin Heidelberg, 2006.
- [14] H. Zhang and J. Hou, “Maintaining Sensing Coverage and Connectivity in Large Sensor Networks,” *Ad Hoc and Sensor Wireless Networks*, vol. 1, no. 1, pp. 89–124, 2005.
- [15] PROFECO, “Revista del consumidor.” Website, Mayo 2013. <http://revistadelconsumidor.gob.mx>.
- [16] OECD, “OECD.” Website, Mayo 2013. <http://www.oecd.org>.
- [17] A. Abdulla, H. Nishiyama, N. Ansari, and N. Kato, “HYMN to Improve the Scalability of Wireless Sensor Networks,” in *Proceedings of the IEEE Conference on Computer Communications Workshops*, (Shanghai, China), pp. 519–524, 2005.
- [18] P. Gupta and P. Kumar, “The Capacity of Wireless Network,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

Bibliografía

- [19] A. Mishra, “Special Issue: Scalability Issues in Wireless Networks Architectures, Protocols and Services,” *Wireless Communications and Mobile Computing*, vol. 4, no. 3, pp. 247–249, 2004.
- [20] A. Agarwal and P. Kumar., “Improved capacity bounds for wireless networks,” *Wireless Communications and Mobile Computing*, vol. 4, no. 3, pp. 251–261, 2004.
- [21] S. Servetto, “Lattice quantization with side information: Codes, asymptotics, and applications in sensor networks,” *IEEE Transaction on Information theory*, vol. 53, no. 2, pp. 714–731, 2007.
- [22] I. Amundson and X. Koutsoukos, “A Survey on Localization for Mobile Wireless Sensor Networks,” in *Proceedings of the 2nd International Conference on Mobile Entity Localization and Tracking in GPS-less Environments*, (Florida, EUA), pp. 235–254, 2009.
- [23] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [24] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, “The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring,” in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, (Colorado, EUA), pp. 29–39, 2008.
- [25] P. Mohan, V. Padmanabhan, and R. Ramjee, “Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones,” in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, (Carolina del Norte, EUA), pp. 323–336, 2008.
- [26] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, E. Shih, H. Balakrishnan, and S. Madden, “CarTel: A Distributed Mobile Sensor Computing System,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, (Colorado, EUA), pp. 125–138, 2006.
- [27] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, “A Survey of Mobile Phone Sensing,” *IEEE Communications*, vol. 48, no. 9, pp. 140–250, 2010.

Bibliografía

- [28] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile Ad hoc Networking*. John Wiley & Sons, 2004.
- [29] Z. Fu, X. Meng, and S. Lu, “How Bad TCP Can Perform in Mobile Ad Hoc Networks,” in *Proceedings of the 7th IEEE Symposium on Computers and Communications*, (Taormina Giardini Naxos, Italia), pp. 298–303, 2002.
- [30] A. Chandrakasan, A. Smith, and W. Heinzelman, “An Application-Specific Protocol Architecture for Wireless Microsensor Networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [31] S. Yoon and C. Shahabi, “The Clustered AGgregation (CAG) Technique Leveraging Spatial and Temporal Correlations in Wireless Sensor Networks,” *ACM Transactions on Sensor Networks*, vol. 3, no. 1, pp. 1–39, 2007.
- [32] M. Vuran and I. Akyildiz, “Spatio Correlation-Based Collaborative Medium Access Control in Wireless Sensor Networks,” *IEEE Transactions on Networking*, vol. 14, no. 2, pp. 316–329, 2006.
- [33] M. M. de Oca, J. Gomez, and M. López-Guerrero, “DISAGREE : Disagreement-based querying in wireless sensor networks,” *Telecommunication Systems*, pp. 1–18, 2013.
- [34] M. Piran, G. Murthy, and G. Babu, “Vehicular Ad hoc and Sensor Networks; Principles and Challenges,” *International Journal of Ad hoc, Sensor & Ubiquitous Computing*, vol. 2, no. 2, pp. 38–49, 2011.
- [35] I. Stojmenovic and S. Olariu, *Data-Centric Protocols for Wireless Sensor Networks*, pp. 417–456. John Wiley & Sons, 2005.
- [36] C. Buratti, A. Conti, D. Dardari, and R. Verdone, “An overview on wireless sensor networks technology and evolution,” *Sensors*, vol. 9, no. 9, pp. 6869–6896, 2009.
- [37] A. Iyer, S. Kulkarniand, V. Mhatre1, and C. Rosenberg, *Wireless Sensor Networks and Applications*, ch. A Taxonomy-based Approach to Design of Large-scale Sensor Networks. Springer, 2008.

- [38] D. Du and P. Pardalo, *Handbook of Combinatorial Optimization*, vol. B, ch. Connected Dominating Set in Sensor Networks and MANETs, pp. 329–370. Springer, 2005.
- [39] K. Akkaya and M. Younis, “A Survey on Routing Protocols for Wireless Sensor Networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.
- [40] J. Al-Karaki and A. Kamal, “Routing Techniques in Wireless Sensor Networks: A Survey,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.
- [41] S. Lee, C. Lee, Y. Cho, and S. Kim, “A New Data Aggregation Algorithm for Clustering Distributed Nodes in Sensor Networks,” in *Universal Multiservice Networks*, vol. 3262 of *Lecture Notes in Computer Science*, pp. 508—520, Springer Berlin Heidelberg, 2004.
- [42] D. Petrovic, R. Shah, K. Ramchandran, and J. Rabaey, “Data Funneling: Routing with Aggregation and Compression for Wireless Sensor Networks,” in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, (Anchorage, AK, EUA), pp. 156–162, 2003.
- [43] M. Watfa, W. Daher, and H. A. Azar, “A Sensor Network Data Aggregation Technique,” *International Journal of Computer Theory and Engineering*, vol. 1, no. 1, pp. 19–26, 2009.
- [44] G. Pottie and W. Kaiser, “Wireless integrated network sensors,” *Communications of the ACM*, vol. 43, no. 5, pp. 551–560, 2000.
- [45] J. Garcia-Macias and J. Gomez, *Sensor Networks and Configuration. Fundamentals, Standards, Platforms, and Applications*, ch. MANET versus WSN, pp. 369–388. Springer Berlin Heidelberg, 2007.
- [46] W. Ye, J. Heidemann, and E. Estrin, “An Energy-Efficient MAC Protocol for Wireless Sensor Networks,” in *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies*, (New York, EUA), pp. 1567–1573, 2002.
- [47] I. Demirkol, C. Ersoy, and F. Alagoz, “Mac protocols for wireless sensor networks: a survey,” *IEEE Communications Magazine*, vol. 44, no. 4, pp. 115–121, 2006.

Bibliografía

- [48] T. V. Dam and K. Langendoen, “An adaptive energy-efficient mac protocol for wireless sensor networks,” in *Proceedings of the 1st International conference on Embedded networked sensor systems*, (Los Angeles, California, EUA), pp. 171–180, 2003.
- [49] J. Polastre, J. Hill, and D. Culler, “Versatile low power media access for wireless sensor networks,” in *Proceedings of the 2nd International conference on Embedded networked sensor systems*, (Baltimore, MD, EUA), pp. 95–107, 2004.
- [50] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, “Energy-efficient collision-free medium access control for wireless sensor networks,” in *Proceedings of the 1st International conference on Embedded networked sensor systems*, (Los Angeles, California, EUA), pp. 181–192, 2003.
- [51] G. Lu, B. Krishnamachari, and C. Raghavendra, “An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks,” in *Proceedings of the 18th International Parallel and Distributed Processing Symposium*, (Los Angeles, California, EUA), pp. 224–230, 2004.
- [52] M. Buettner, G. Yee, E. Anderson, and R. Han, “X-mac: a short preamble MAC protocol for duty-cycled wireless sensor networks,” in *Proceedings of the 4th International conference on Embedded networked sensor systems*, (Boulder, Colorado, EUA), pp. 307–320, 2006.
- [53] H. Oh and P. Van, “Design and implementation of a mac protocol for timely and reliable delivery of command and data in dynamic wireless sensor networks,” *Sensors*, vol. 13, no. 10, pp. 13228–13257, 2013.
- [54] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, “Directed Diffusion for Wireless Sensor Networking,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.
- [55] J. Kulik, W. Heinzelman, and H. Balakrishnan, “Negotiation-based protocols for disseminating information in wireless sensor networks,” *Wireless Networks*, vol. 8, no. 2, pp. 169–185, 2002.

- [56] D. Braginsky and D. Estrin, “Rumor Routing Algorithm for Sensor Networks,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, (Shenzhen, China), pp. 22–31, 2002.
- [57] Y. Yao and J. Gehrke, “The Cougar Approach to In-Network Query Processing in Sensor Networks,” *ACM SIGMOD Record*, vol. 31, no. 3, pp. 9–18, 2002.
- [58] N. Sadagopan, B. Krishnamachari, and A. Helmy, “Active query forwarding in sensor networks,” *Ad Hoc Networks*, vol. 3, no. 1, pp. 91—113, 2005.
- [59] A. Manjeshwar and D. Agrawal, “TEEN : A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks,” in *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, (San Francisco, EUA), pp. 2009–2015, 2001.
- [60] M. Moazeni and A. Vahdatpour, “HEAP: A hierarchical energy aware protocol for routing and aggregation in sensor networks,” in *Proceedings of the 3rd International Conference on Wireless Internet*, (Bruselas, Belgica), pp. 1–7, 2007.
- [61] C. Barenco, R. Gonzalez, N. Cardenas, and L. Garcia, “A proposal of a Wireless Sensor Network Routing Protocol,” *Telecommunication Systems*, vol. 38, no. 1, pp. 61–68, 2008.
- [62] D. Yang, X. Li, R. Sawhney, and X. Wang, “Geographic and energy aware routing in wireless sensor networks,” *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 4, no. 2, pp. 61–70, 2009.
- [63] H. Tian, J. Stankovic, L. Chenyang, and T. Abdelzaher, “SPEED: a stateless protocol for real-time communication in sensor networks,” in *Proceedings of the 23rd International Conference on Distributed Computing Systems*, pp. 46–55, 2003.
- [64] Z. Haas, J. Halpern, and L. Li, “Gossip-based Ad hoc Routing,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 479–491, 2006.
- [65] S. Hedetniemi and A. Liestman, “A Survey of Gossiping and Broadcasting in Communication Networks,” *Networks*, vol. 18, no. 4, pp. 319–349, 1988.

- [66] J. Wang, Z. Zhang, F. Xia, W. Yuan, and S. Lee, “An energy efficient stable election-based routing algorithm for wireless sensor networks,” *Sensors*, vol. 13, no. 11, pp. 14301–14320, 2013.
- [67] S. Zafar, “A survey of transport layer protocols for wireless sensor networks,” *International Journal of Computer Applications*, vol. 33, no. 1, pp. 44–50, 2011.
- [68] C. Wan, S. Eisenman, and A. Campbell, “CODA: Congestion detection and avoidance in sensor networks,” in *Proceedings of the First International Conference on Embedded Networked Sensor Systems*, (Los Angeles, California, EUA), pp. 266—279, 2003.
- [69] O. Akan and I. Akyildiz, “Event-to-sink reliable transport in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1003–1016, 2015.
- [70] B. Sharma and C. Aseri, “A comparative analysis of reliable and congestion-aware transport layer protocols for wireless sensor networks,” *ISRN Sensor Networks*, vol. 2012, pp. 1–14, 2012.
- [71] P. Hurni, U. Burgi, M. Anwander, and T. Brau, “TCP performance optimizations for wireless sensor networks,” in *Proceedings of the 9th European Conference on Wireless Sensor Networks*, (Trento, Italia), pp. 17–32, 2009.
- [72] L. Jason, F. Perrone, D. Nicol, M. Liljenstam, C. Elliot, and D. Pearson, “Simulation modeling of large-scale ad-hoc sensor networks,” in *Proceedings of the European Simulation Interoperability Workshop*, pp. 1–10, 2001.
- [73] D. Networks, “Dust Networks.” Website, Julio 2013. <http://www.dust-inc.com>.
- [74] P. Juang, H. Oki, Y. Wang, M. Martonosi, and L. Peh, “Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet,” *SIGOPS Oper. Syst. Rev.*, vol. 35, no. 5, pp. 96–107, 2002.
- [75] A. Beaufour, M. Leopold, and P. Bonnet, “Smart-tag based data dissemination,” in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, (Atlanta, Georgia, EUA), pp. 68–77, 2002.

- [76] D. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole, “Research challenges in environmental observation and forecasting systems,” in *Proceedings of the 6th International Conference on Mobile Computing and Networking*, (Boston, EUA), pp. 292–299, 2000.
- [77] A. Baptista, “CMOP.” Website, Julio 2013. <http://www.ccalmr.ogi.edu>.
- [78] K. Martinez, A. Riddoch, J. Hart, and R. Ong, “A sensor network for glaciers,” in *Intelligent Spaces*, pp. 125–138, Springer Berlin Heidelberg, 2006.
- [79] E. Yoneki and J. Bacon, “A survey of Wireless Sensor Network technologies: research trends and middleware’s role,” tech. rep., University of Cambridge. Computer Laboratory, 2006.
- [80] L. Schwiebert, S. Gupta, and J. Weinmann, “Research challenges in wireless networks of biomedical sensors,” in *Proceedings of the 7th International Conference on Mobile Computing and Networking*, (New York, EUA), pp. 151–165, 2001.
- [81] C. Kidd, R. Orr, G. Abowd, C. Atkeson, I. Essa, B. MacIntyre, E. Mynatt, T. Starner, and W. Newstetter, “The Aware Home: A Living Laboratory for Ubiquitous Computing Research,” in *Proceedings of the 2nd International Workshop on Cooperative Buildings*, (Pittsburgh, PA, EUA), pp. 191–198, 1999.
- [82] D. Snoonian, “Smart buildings,” *IEEE Spectrum*, vol. 40, no. 8, pp. 18–23, 2003.
- [83] F. Losilla, A. Garcia-Sanchez, F. Garcia-Sanchez, J. Garcia-Haro, and Z. Haas, “A Comprehensive Approach to WSN-Based ITS Applications: A Survey,” *Sensors*, vol. 11, no. 11, pp. 10220–10265, 2011.
- [84] L. Holmquist, H. Gellersen, G. Kortuem, S. Antifakos, F. Michahelles, B. Schiele, M. Beigl, and R. Maze, “Building intelligent environments with smart-its,” *IEEE Computer Graphics and Applications*, vol. 24, no. 1, pp. 56–64, 2004.
- [85] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols and Applications*. John Wiley & Sons, 2007.
- [86] F. Zhao and L. Guibas, *Wireless Sensor Networks An Information Processing Approach*. Morgan Kaufmann, 2004.

- [87] S. Choi, Y. Kim, J. Kim, and H. Park, “Field monitoring of column shortenings in a high-rise building during construction,” *Sensors*, vol. 13, no. 11, pp. 14321–14338, 2013.
- [88] E. Oliver, “A Survey of Platforms for Mobile Networks Research,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 12, no. 4, pp. 56–63, 2009.
- [89] R. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.
- [90] P. Dutta, P. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, “Demo abstract: Common sense: Participatory urban sensing using a network of handheld air quality monitors,” in *Proceedings of the 7th on Embedded Networked Sensor Systems*, (Los Angeles, California, EUA), pp. 349—350, 2009.
- [91] W. Chen and J. Chou, *Data Gathering and Fusion in Sensor Networks*, pp. 493–526. John Wiley & Sons, 2005.
- [92] R. Rajagopalan and P. Varshney, “Data aggregation techniques in sensor networks: A survey,” *IEEE Communications Surveys Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.
- [93] M. Francesco, S. Das, and G. Anastasi, “Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey,” *ACM Transactions on Sensor Networks*, vol. 8, no. 1, pp. 1–31, 2011.
- [94] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. Eisenman, X. Zheng, and A. Campbell, “Sensing meets Mobile Social Networks: The Design, Implementation, and Evaluation of the CenceMe Application,” in *Proceedings of the 6th International Conference on Embedded Networked Sensor Systems*, (Carolina del Norte, EUA), pp. 337–350, 2008.
- [95] J. Heidemann, F. Silva, and D. Estrin., “Matching data dissemination algorithms to application requirements,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, (Los Angeles, California, EUA), pp. 218–229, 2003.

- [96] X. Li, Y. Kim, R. Govindan, and W. Hong, “Multi-dimensional range queries in sensor networks,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, (Los Angeles, California, EUA), pp. 63–75, 2003.
- [97] H. Gupta, V. Navda, S. Das, and V. Chowdhary, “Efficient Gathering of Correlated Data in Sensor Networks,” *ACM Transactions on Sensor Networks*, vol. 4, no. 1, pp. 1–31, 2008.
- [98] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, “The Broadcast Storm Problem in a Mobile Ad-hoc Network,” in *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, (Seattle, EUA), pp. 151–162, 1999.
- [99] K. Lee, J. Lee, H. Lee, and Y. Shin, “A Density and Distance based Cluster Head Selection algorithm in Sensor Networks,” in *Proceedings of the 12th International Conference on Advanced Communication Technology*, (Phoenix, EUA), pp. 162–165, 2010.
- [100] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.
- [101] H. Kour and A. K. Sharma, “Hybrid Energy Efficient Distributed Protocol for Heterogeneous Wireless Sensor Network,” *International Journal of Computer Applications*, vol. 4, no. 5, pp. 37–41, 2010.
- [102] Y. Jin, F. Chen, G. Che, and W. Hu, “Energy-efficient Data Collection Protocol for Wireless Sensor Network based on Tree,” in *Proceedings of the Asia-Pacific Conference on Wearable Computing Systems*, (Shenzhen, China), pp. 82–85, 2010.
- [103] N. Israr and I. Awan, “Coverage based inter cluster communication for load balancing in heterogeneous Wireless Sensor Networks,” *Telecommunication Systems*, vol. 38, no. 3, pp. 121–132, 2008.
- [104] X. Guan, L. Guan, X. Wang, and T. Ohtsuki, “A new load balancing and data collection algorithm for energy saving in Wireless Sensor Networks,” *Telecommunication Systems*, vol. 45, no. 4, pp. 313–322, 2010.

- [105] A. Manjeshwar and D. Agrawal, “APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks,” in *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, (Florida, EUA), pp. 195–202, 2002.
- [106] Y. Wang, Y. Hsieh, and Y. Tseng, “Multiresolution Spatial and Temporal Coding in a Wireless Sensor Networks for Long-Term Monitoring Applications,” *IEEE Transactions on Computers*, vol. 58, no. 6, pp. 827–838, 2009.
- [107] C. Sadler and M. Martonosi, “Data compression algorithms for energy-constrained devices in delay tolerant networks,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, (New York, EUA), pp. 265–278, 2006.
- [108] K. Sayood, *Introduction to Data Compression*. Morgan Kaufmann Series, 2012.
- [109] D. Ganesan, D. Estrin, and J. Heidemann, “DIMENSIONS: Why do we need a new data handling architecture for Sensor Networks?,” *ACM SIGCOMM Computer Communications*, vol. 33, no. 1, pp. 143–148, 2003.
- [110] A. Graps, “An Introduction to Wavelets,” *IEEE Computational Science and Engineering*, vol. 2, no. 2, pp. 50–61, 1995.
- [111] D. Chu, A. Deshpande, J. Hellerstein, and W. Hong, “Approximate Data Collection in Sensor Networks using Probabilistic Models,” in *Proceedings of the 22nd International Conference on Data Engineering*, (Atlanta, EUA), pp. 48–48, 2006.
- [112] M. Vuran, C. Akan, and I. Akyildiz, “Spatio-temporal Correlation: Theory and Applications for Wireless Sensor Networks,” *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 45, no. 3, pp. 245–259, 2004.
- [113] J. Chen, Y. Guan, and U. Pooch, “A Spatial-based Multi-resolution Data Dissemination Scheme for Wireless Sensor Networks,” in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, (Denver, EUA), pp. 245–253, 2005.

- [114] S. Kondo, A. Kanzaki, T. Hara, and S. Nishio, “Energy-Efficient Data Gathering using Sleep Scheduling and Spatial Correlation based on Data Distribution in Wireless Sensor Networks,” in *Proceedings of the 14th International Conference on Network-Based Information Systems*, (Tirana, Albania), pp. 194–201, 2011.
- [115] Y. Iima, A. Kanzaki, T. Hara, and S. Nishio, “Overhearing based Data Transmission Reduction for Periodical Data Gathering in Wireless Sensor Networks,” in *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems*, (Fukuoka, Japan), pp. 1048–1053, 2009.
- [116] F. Dai and J. Wu, “On constructing k-connected k-dominating set in wireless ad hoc and sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 66, no. 7, pp. 947–958, 2006.
- [117] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press, McGrawHill, 2001.
- [118] A. Roxin, J. Gaber, M. Wack, and A. Nait-Sidi-Moh, “Survey of Wireless Geolocation Techniques,” in *IEEE Globecom Workshops*, (Washington, DC, EUA), pp. 1–9, 2007.
- [119] Y. Wang, J. Gao, and J. Mitchell, “Boundary Recognition in Sensor Networks by Topological Methods,” in *Proceedings of the 12th International Conference on Mobile Computing and Networking*, (New York, EUA), pp. 122–133, 2006.
- [120] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, “Network Correlated Data Gathering with Explicit Communication: NP-Completeness and Algorithms,” *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, pp. 41–54, 2006.
- [121] C. Zhang, B. Wang, S. Fang, and J. Zheng, “Spatial Data Correlation Based Clustering Algorithms for Wireless Sensor Networks,” in *Proceedings of the 3rd International Conference on Innovative Computing Information and Control*, (Dalian, Liaoning, China), p. 593, 2008.
- [122] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “TinyDB: An Acquisitional Query Processing System for Sensor Networks,” *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 122–173, 2005.

- [123] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, “TAG: A Tiny AGgregation service for Ad-hoc Sensor Networks,” in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, (New York, EUA), pp. 131–146, 2002.
- [124] H. Miller, “Tobler’s First Law and spatial analysis,” *Annals of the Association of American Geographers*, vol. 94, no. 2, pp. 284—289, 2004.
- [125] O. Riva and C. Borcea, “The Urbanet revolution Sensor power to the people,” *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 41–49, 2007.
- [126] N. Lane, S. Eisenman, M. Musolesi, E. Miluzzo, and A. Campbell, “Urban sensing: Opportunistic or participatory?,” in *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications*, (Napa Valley, California, EUA), pp. 11–16, 2008.
- [127] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. Srivastava, “Participatory Sensing,” in *Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications*, pp. 117–134, 2006.
- [128] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, and R. Peterson, “People-Centric Urban Sensing,” in *Proceedings of the 2nd Annual International Workshop on Wireless Internet*, (Boston, Massachusetts), pp. 1–14, 2006.
- [129] A. Parker, S. Reddy, T. Schmid, K. Chang, G. Saurabh, M. Srivastava, M. Hansen, J. Burke, D. Estrin, M. Allman, and V. Paxson, “Network System Challenges in Selective Sharing and Verification for Personal, Social, and Urban-Scale Sensing Applications,” in *In HotNets*, pp. 37–42, 2006.
- [130] I. Carreras, D. Miorandi, A. Tamilin, E. Ssebaggala, and N. Conci, “Crowdsensing: why context matters,” in *Proceedings of the 11st IEEE International Conference on Pervasive Computing*, (San Diego, EUA), pp. 368–371, 2013.
- [131] B. Mihajlov and M. Bogdanoski, “Overview and Analysis of the Performance of ZigBee-based wireless Sensor Networks,” *International Journal of Computer Applications*, vol. 29, no. 12, pp. 28–35, 2011.

- [132] M. Zuckerberg, E. Saverin, C. Hughes, and D. Moskovitz, “Facebook.” Website, Julio 2013. <http://www.facebook.com>.
- [133] M. Mun, S. Reddy, K. Shilton, N. Yau, P. Boday, J. Burke, D. Estrin, M. Hansen, E. Howard, and R. West, “PEIR, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research,” in *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, (Krakow, Polonia), pp. 55–68, 2009.
- [134] M. Bilandzic, M. Banholzer, D. Peev, V. Georgiev, F. Balagtas-Fernandez, and A. D. Luca, “Laermometer - a mobile noise mapping application,” in *Proceedings of the 5th Nordic Conference on Human-computer Interaction: Building Bridges*, (Lund, Suecia), pp. 415–418, 2008.
- [135] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Toledo, and J. Eriksson, “VTrack: Accurate, Energy-Aware Road Traffic Delay Estimation Using Mobile Phones,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor System*, (Berkeley, California, EUA), pp. 85–98, 2009.
- [136] Trafficsense, “Traffic Sense.” Website, Enero 2012. <http://www.trafficsense.co.uk>.
- [137] T. Das, P. Mohan, V. Padmanabhan, R. Ramjee, and A. Sharma, “PRISM: Platform for Remote Sensing using Smartphones,” in *Proceedings of 8th ACM Conference on Mobile Systems, Applications, and Services*, (San Francisco, EUA), pp. 63–76, 2010.
- [138] S. Gaonkar, L. Jack, C. Rommit, C. Landon, and A. Schmidt, “Micro-Blog: Sharing and Querying Content through Mobile Phones and Social Participation,” in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, (Breckenridge, CO, EUA), pp. 174–186, 2008.
- [139] N. Thepvilojanapong, S. Konomi, Y. Tobe, Y. Ohta, M. Iwai, and K. Sezaki, “Opportunistic collaboration in participatory sensing environments,” in *Proceedings of the 5th ACM International Workshop on Mobility in the Evolving Internet Architecture*, (Chicago, Illinois, EUA), pp. 39–44, 2010.

Bibliografía

- [140] N. Ramanathan, S. Reddy, J. Burke, and D. Estrin, “Participatory sensing for project surya,” in *Workshop on Sensing on Everyday Mobile Phones in Support of Participatory Research at SenSys*, (Sydney, Australia), pp. 1–4, 2007.
- [141] A. Beach, M. Gartrell, S. Akkala, J. Elston, J. Kelley, K. Nishimoto, B. Ray, S. Razgulin, K. Sundaresan, B. Surendar, and M. T. R. Han, “WhozThat? evolving an ecosystem for context-aware mobile social networks,” *IEEE Communications*, vol. 22, no. 4, pp. 50–55, 2008.
- [142] H. Lu, N. Lane, S. Eisenman, and A. Campbell, “Bubble-Sensing: Binding Sensing Tasks to the Physical World,” *Pervasive and Mobile Computing*, vol. 6, no. 1, pp. 58–71, 2010.
- [143] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, “TrafficView: A Scalable Traffic Monitoring System,” in *Proceedings of International Conference on Mobile Data Management*, (Berkeley, EUA), pp. 13–26, 2004.
- [144] P. Zhou, T. Nadeem, P. Kang, C. Borcea, and L. Iftode, “EZCab: A Cab Booking Application Using Short-Range Wireless Communication,” in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, (Kauai, Hawaii), pp. 27–38, 2005.
- [145] A. Kansal, M. Goraczko, and F. Zhao, “Building a Sensor Network of Mobile Phones,” in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*, (Massachusetts, EUA), pp. 547–548, 2007.
- [146] S. E. N. Lane, H. Lu and A. Campbell, “Cooperative techniques supporting sensor-based people-centric inferencing,” in *Proceedings of the 6th International Conference on Pervasive Computing*, (Sydney, Australia), pp. 75–92, 2008.
- [147] Y. Liang, *Introduction to Java Programming, Comprehensive Version*. 2012: Pearson.
- [148] R. Meier, *Professional Android 4 Application Development*. 2012: John Wiley & Sons.
- [149] G. Millette and A. Stroud, *Professional Android Sensor Programming*. 2012: John Wiley & Sons.

Bibliografía

- [150] R. Tyan, G. O'Hare, D. Marsh, and D. O'Kane, "Interpolation for Wireless Sensor Network Coverage," in *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors*, (Sydney, Australia), pp. 123–131, 2005.
- [151] R. Schafer and R. Hammell, "Spatial Interpolation for a Wireless Sensor Network of Chemical Point Detectors: Preliminary Results," in *Proceedings of the Conference on Information Systems Applied Research*, (Washington, DC, EUA), pp. 1–8, 2009.
- [152] C. Volosencu and D. Curiac, "Interpolation Techniques for Spatial Distributed System Identification Using Wireless Sensor Networks," in *Proceedings of the 10th WSEAS International Conference on Automation & Information*, (Stevens Point, Wisconsin, USA), pp. 373–376, 2009.
- [153] R. Gonzalez and R. Woods, *Digital Image Processing*. Pearson Prentice Hall, 2008.
- [154] M. Project, "The CMU Monarch Project." Website, Marzo 2012. [http://http://www.monarch.cs.cmu.edu](http://www.monarch.cs.cmu.edu).
- [155] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.
- [156] L. Kyunghan, H. Seongik, J. Seong, R. Injong, and C. Song, "Slaw: A mobility model for human walks," in *Proceedings of INFOCOM 2009*, (Rio de Janeiro, Brasil), pp. 855–863, 2009.
- [157] R. Pushpa, A. Vallimayil, and V. Dhulipala, "Impact of mobility models on mobile sensor networks," in *Proceedings of the 3rd International Conference on Electronics Computer Tecnology*, (Kanyakumari, India), pp. 102–106, 2011.
- [158] J. Markoulidakis, G. Lyberopoulos, D. Tsirkas, and E. Sykas, "Mobility modeling in third generation mobile telecommunication systems," *IEEE Personal Communications*, vol. 4, no. 4, pp. 41–56, 1997.
- [159] T. Camp, J. B. amd, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2002.*, vol. 2, no. 1, pp. 483–502, 2002.

Bibliografía

- [160] W. Lee, *Beginning Android Application Development*. John Wiley & Sons, 2011.
- [161] L. Electronics, “LG.” Website, Mayo 2013. <http://www.lg.com>.
- [162] Samsung, “Samsung.” Website, Mayo 2013. <http://www.samsung.com>.
- [163] Motorola, “Motorola.” Website, Mayo 2013. <http://www.razri.com>.
- [164] K. Shilton, “Four billion little brothers?: Privacy, mobile phones, and ubiquitous data collection,” *Queue*, vol. 7, no. 7, pp. 40–47, 2009.