

Desarrollo e implementación de un simulador para el seguimiento de objetos en redes de sensores

Ana Karen Hernández-Vázquez, Alexis Jair López-Gutiérrez,
Perfecto Malaquías Quintero-Flores

Instituto Tecnológico de México,
Instituto Tecnológico de Apizaco,
Departamento de Estudios de Posgrado,
México

{anakarenhv26,alexistegno,parfait.phd}@gmail.com

Resumen. El seguimiento de objetos tiene varias aplicaciones en la vida real, como la vigilancia en redes de vehículos, monitoreo de vida silvestre, propagación de incendios, observación de pacientes, localización, etc. Las redes de sensores consisten en un conjunto de cientos o miles de dispositivos autónomos, los cuales son llamados nodos sensores, estos nodos están distribuidos físicamente en áreas geográficas para poder monitorear condiciones físicas o ambientales, con una capacidad de almacenamiento y comunicación de datos en una red de forma inalámbrica a corta distancia. Estas aplicaciones deben de recopilar periódicamente los datos sensoriales y usarlos para reconstruir el estado general del área supervisada mediante la agregación de datos. Este trabajo presenta el seguimiento de objetos mediante: 1.- Estructura Jerárquica de Agrupación de sensores para red de sensores, 2.- Seguimiento de ubicación basado en agrupación para redes de sensores de objetos. Los resultados de la simulación muestran la disminución de sobrecarga de comunicación sin disminuir la precisión del seguimiento de objetos y se lo logra la extensión de la vida útil de la red.

Palabras clave: Redes de sensores, seguimiento de objetos, eficiencia energética, vida útil de la red.

Development and Implementation of a Simulator for Tracking Objects in Sensor Networks

Abstract. Object tracking has many applications in real life, such as vehicle network surveillance, wildlife monitoring, fire spread, patient observation, location, etc. Sensor networks consist of a set of hundreds or thousands of autonomous devices, which are called sensor nodes, these nodes are physically distributed in geographic areas to be able to monitor physical or environmental conditions, with a capacity for data storage

and communication in a network wirelessly at close range. These applications should periodically collect sensory data and use it to reconstruct the overall condition of the monitored area through data aggregation. This work presents the tracking of objects through: 1.- Hierarchical Structure of Sensor grouping for sensor network, 2.- Location tracking based on grouping for sensor networks of object tracking. The simulation results show the decrease in communication overhead without decreasing the accuracy of the object tracking and the extension of the useful life of the network is achieved.

Keywords: Sensor networks, object tracking, energy efficiency, network lifetime.

1. Introducción

La visión futurista que Mark Weiser describió en su artículo *The Computer for the 21st Century* [13] es una realidad, se describen entornos saturados de elementos con capacidades de computo y comunicación, totalmente integrados en la vida de las personas y que proporcionan información asociada a las necesidades y al entorno en el que se encuentren en cada momento.

Las redes de sensores presentan una amplia variedad de implementaciones que son utilizadas como aplicaciones, en donde el nodo sensor o varios sensores pueden ser capaces de modelar el estado de un entorno en particular. Entre las múltiples aplicaciones de las redes de sensores se pueden encontrar procesos como monitoreo, vigilancia, seguridad y tácticas de vigilancia, control de inventarios, seguimiento de objetos o personas, atención médica, etc. El desarrollo de este tipo de redes plantea nuevos problemas, pero a su vez acerca más a la realidad escenarios que antes se pensaron en el plano de ciencia ficción.

El resto de este artículo está organizado de la siguiente manera, en la sección 2 se presentan los antecedentes. El trabajo relacionado es presentado en la sección 3. Desarrollo metodológico forma parte de la sección 4. El desarrollo de trabajo experimental e interpretación de resultados obtenidos se presenta en la sección 5. La sección 6 concluye este trabajo.

2. Trabajo relacionado

El seguimiento de objetos de redes de sensores ha llamado la atención de los investigadores debido a su amplia aplicación, la mayoría de estos estudios no pueden abordar el equilibrio entre la eficiencia energética y la precisión del seguimiento. Como parte de la implementación de algoritmos que mejoren el seguimiento de objetos en ámbitos internos y externos, se han publicado un número significativo de propuestas con el enfoque de seguimiento de objetos en redes de sensores. Existe una gran demanda de seguimiento de ubicación de objetos en movimiento desde diversos servicios basados en ubicación [15]. Una de las aplicaciones más desafiantes de tal tecnología es el seguimiento de objetos, objetivos y/o eventos en movimiento.

En este tipo de redes de sensores, la energía es un problema clave, ya que afecta directamente la vida útil de la red. La energía de las redes se puede conservar cambiando el nodo de un sensor a modo de suspensión cuando no hay ningún objeto en el rango de detección del nodo y activando un nodo del sensor cuando un objeto debe ingresar al rango de detección. Es probable que la ubicación del objeto, la hora de llegada y la ruta oculten algunas reglas de asociación útiles, que se pueden extraer mediante la aplicación del algoritmo de minería de datos adecuado. Es por lo que los autores en [10] han propuesto un esquema de seguimiento de objetos para redes de sensores de seguimiento de objetos (OTSN por sus siglas en inglés) utilizando el enfoque de minería de datos.

En este esquema se ha mejorado el algoritmo Apriori para las reglas de asociación minera y se ha aplicado a *OTSN*, este algoritmo se aplica a la información de movimiento anterior del objeto y se excavan reglas de asociación útiles, que luego se utilizan para predecir la siguiente ubicación del objeto con mayor precisión y aumentar la vida útil de la red. El filtro de Kalman [12], el filtro de partículas y la minería de datos, son los modelos más utilizados para predecir la siguiente ubicación del objeto, sin embargo, estos modelos no toman en consideración la durabilidad de la vida de la red de sensores, es por ellos que este trabajo ofrece un enfoque diferente, donde la vida útil de la red y la eficiencia energética son dos de los mayores puntos a atacar.

Hawbani et al [6] proponen un nuevo modelo de seguimiento, denominado *Seguimiento de ubicación basado en agrupación en redes de sensores de seguimiento de objetos*. (*GLT* por sus siglas en inglés) esta basado en la estructura jerárquica de agrupación (*GHS* por sus siglas en inglés) [7], en *GHS* los nodos se dividen en grupos de acuerdo con su región de cobertura máxima, tal que cada grupo contiene un número de nodos y un número de líderes. *GLT* esta conformado por dos niveles, el primer nivel denominado árbol de notificación (*NT* por sus siglas en inglés), mejora el mecanismo de activación, el mecanismo de limpieza de datos y el mecanismo de balance de energía. En *GLT* no hay necesidad de un mecanismo de recuperación, ya que *NT* no se basa en predicciones. Por otro lado, el segundo nivel, árbol de expansión jerárquica, soporta el mecanismo de reporte de datos y el mecanismo de prolongación de vida útil. El árbol de notificación y el árbol de expansión jerárquica están diseñados para trabajar juntos y así poder lograr su mejor desempeño en términos de vida útil de la red, precisión en el seguimiento, reducción de tráfico y reducción de retardo de transmisión. *GLT* garantiza un alto rendimiento con el soporte del árbol (*NT*) y el árbol (*HST*, por sus siglas en inglés) para agregar e informar. Los autores en [7] proporcionan una estructura jerárquica de agrupación (*GHS*, por sus siglas en inglés) para dividir los nodos en la red de sensores inalámbricos en grupos para ayudar a la computación y comunicación de la colaboración dinámica y distribuida del sistema.

Para obtener un trabajo de colaboración estable entre los sensores, los nodos deben poder organizarse en estructuras (estructura jerárquica) de modo que los objetivos de la red se alcancen utilizando el costo mínimo de comunicación.

La idea es dividir los nodos según sus regiones geográficas máximas cubiertas de manera que cada grupo contenga una serie de nodos y una serie de líderes. El impacto de este modelo se ve reflejado en (i) enrutamiento de datos y ahorro de energía, (ii) seguimiento de objetos y balance de energía y (iii) evitar la redundancia de datos. Para evaluar el rendimiento de la estructura los autores implementaron un enrutamiento basado en grupos y un seguimiento de objetos basado en grupos. La estructura de agrupación muestra un buen rendimiento en el consumo de energía y la disipación de energía durante el enrutamiento de los objetos y que se generan pocos datos redundantes durante el seguimiento de objetos. En el trabajo [3] los autores presentan un análisis y comparación de diferentes algoritmos para la localización de sensores, para la evaluación de los algoritmos, se simuló una red inalámbrica de sensores en MATLAB con un área delimitada de 100 m x 100 m en donde se distribuyeron aleatoriamente 100 sensores con rango parcial para su localización.

Una vez que cada sensor en la red estimó su posición inicial, se utilizó dicho conjunto de posiciones iniciales como punto de inicio para los diferentes algoritmos analizados. Los resultados obtenidos muestran que los algoritmos iterativos optimizados son más eficientes en precisión y número de iteraciones. Entre los algoritmos de localización analizados se encuentran: LeastSquare, Min-Max, Spatially-Constrained Local Problems, Push-Pull Estimator y Levenberg-Marquardt. TS Chen et al. [1] proponen un método de seguimiento de objetos distribuidos utilizando la estructura de la red de polígonos convexos, llamados caras, los nodos en las caras cooperan para encontrar las trayectorias de un objeto y luego estas trayectorias se utilizan para predecir el movimiento de los objetos. El método propuesto basado en la construcción del árbol de trayectoria puede reducir tanto el espacio de almacenamiento de las trayectorias recolectadas como el tiempo dedicado al análisis de predicción de trayectorias. Las simulaciones muestran que el método propuesto puede reducir el consumo de energía de los nodos y hacer una predicción de la dirección de movimiento de los nodos con precisión.

3. Desarrollo metodológico

En la literatura se han propuesto muchos protocolos para reducir el consumo de energía en las redes de sensores para el seguimiento de objetos. Además, en las redes de sensores, los modelos de agrupación, modelo de agregación de datos, modelo de activación de nodos y modelo de informe de datos juegan un papel esencial que podría maximizar la vida útil de la red [4, 11]. Con estas motivaciones, la idea que se presenta en este trabajo es de implementar GLT [6] y GHS [7] para el seguimiento de objetos, objetivos y/o eventos.

3.1. Modelo estructural

GHS y GLT: La idea principal es dividir los nodos de la red en grupos según sus regiones geográficas máximas cubiertas de tal manera que cada grupo contenga un número de nodos y un número de líderes.

Por otro lado, *GLT* es un modelo para detectar movimientos de un objetivo utilizando la Estructura Jerárquica de agrupación [7]. Este modelo de agrupación tiene varias ventajas que hace que las redes de sensores estén libres de estrés en muchos aspectos, como el enrutamiento de datos y evitar la redundancia de datos. A continuación, se proporciona un algoritmo de agrupación distribuido (algoritmo 1, que se ejecuta dentro de la unidad de procesamiento del nodo sensor).

Se compone de dos niveles. El primer nivel, la zona de activación (Árbol de notificaciones/NT), mejora el mecanismo de activación, el mecanismo de limpieza de datos y el mecanismo de equilibrio de energía. El segundo nivel llamado Árbol de expansión jerárquica, es compatible con el mecanismo de informe de datos y el mecanismo de prolongación de la vida útil.

Definición 1 (*lista de vecinos*). Para cada nodo $s_a \in S$ su lista de vecinos se define como un conjunto de nodos $nl(S_a) = \{S_b | S_b \in S \text{ y } \sqrt{(X_{S_a} - X_{S_b})^2 + (Y_{S_a} - Y_{S_b})^2} < r_a + r_b\}$ donde r_a y r_b son los rangos de detección para S_a y S_b respectivamente; S es el conjunto para todos los nodos sensores en la red. $nl(S) = nl(S_i \text{ — } S_i \in S)$ es una matriz de listas de vecinos.

Definición 2 (*grupo*). Un conjunto de nodos, denotado como $G^k = \{s_0 \dots, s_j\} - \forall s_i \in S$, se dice que es un grupo de tamaño k (es decir, contiene k nodos) si y solo si $G^k = \{s_0 \dots s_k\}$ ocurre exactamente k veces como un subconjunto en la matriz $nl(S)$. La matriz de todos los grupos en la red se denota por G^* .

Definición 3 (*grupos asociados*). Los grupos asociados para s_j son definidos como un conjunto de grupos. $G^*(S_j) = \{G^k | G^k \in G^* \text{ y } S_j \in G^k\}$.

Modelo de seguimiento de objetos: Se definen tres estados para cada nodo: activo, inactivo y en reposo. En estado activo, el nodo recopila los datos sensoriales y los reporta al nodo de agregación, mientras que en estado inactivo, el nodo se prepara para rastrear le objeto y espera una activación el nodo proxy seleccionado. En el estado de reposo no se pueden recopilar datos sensoriales. Los nodos soportan dos tipos de mensajes:

- Insertar paquetes de control.
- Borrar paquetes de control.

Tabla 1. Símbolos usados en el algoritmo de agrupación.

Simbolo	Significado
S	Un conjunto de sensores.
D	Distancia euclidiana
V_i	<p>La lista de nodos vecinos de s_i.</p> <p>Aquí llamamos a V el vector del sensor.</p> <p>V_i Es el número de sensores dentro de V_i.</p> <p>$D = \sqrt{(x_{s_i} - x_{s_j})^2 + (y_{s_i} - y_{s_j})^2}$</p> <p>$\forall s_j \in S$ hacer :</p> <p>{</p> <p>$D = \sqrt{(x_{s_i} - x_{s_j})^2 + (y_{s_i} - y_{s_j})^2}$</p> <p>$\forall s_j \in S$ hacer :</p> <p>{</p> <p>si($r_i + r_j < D$) hacer :</p> <p>$V_i.add(s_j)$;</p> <p>}</p>
N_i	<p>La lista de vectores vecinos para s_i</p> <p>$\forall s_j \in V_i$ hacer</p> <p>Encuentre V_j y luego agrega V_j a N_i llamando a:</p> <p>$N_i.add(V_j)$.</p> <p>Esto es encontrar el vector para cada nodo sensor s_j tal que $s_j \in V_i$ $i \neq j$.</p> <p>El número de vectores en N_i es denotado por N_i</p>
F_i	<p>La lista de vectores vecinos filtrados para s_i. El proceso de filtrado se ejecuta como:</p> <p>$\forall V_a \in N_i$ hacer :</p> <p>{</p> <p>$V_x \leftarrow null$;</p> <p>$\forall s_c \in V_a$ hacer :</p> <p>{</p> <p>si($s_c \in V_i$)hacer :</p> <p>{</p> <p>$V_x.add(s_c)$;</p> <p>}</p> <p>}</p> <p>$F_i.add(V_x)$</p> <p>}</p> <p>Es decir, el sensor en N_i se agregará a F_i si y solo si pertenecen a V_i de lo contrario serán ignorados. F_i es el número de vectores dentro de F_i.</p>
$G^*(s_j)$	Los grupos asociados de s_i , de modo que $G^*(s_j) = \{G^x \mid G^x \in G^* \text{ y } s_j \in G^x\}$.
$F_i[n]$	Obtiene el vector filtrado con el índice n .

Algoritmo 1 Algoritmo de agrupación.

Entrada: s_i, S

Salida: $G^*(s_i)$ los grupos asociados de s_i

```

1:  $V_i \leftarrow s_i$ . Vector
2: Para (int  $k \leftarrow 0$  ;  $k \leq |F_i|$ ;  $k++$ ) Hacer
3:    $V_k \leftarrow F_i[k]$ ;
4:    $s_k \leftarrow V_i[k]$ ;
5:   Si ( $|V_k|=2$ ) entonces
6:      $G^k \leftarrow V_k$ 
7:      $G^*(s_i) += G^k$ 
8:   Si no
9:     Para (int  $m \leftarrow k + 1$ ;  $m < |F_i|$ ;  $m++$ ) Hacer
10:       $V_m \leftarrow F_i[m]$ ;
11:      Si  $S_k \in V_m$  entonces
12:         $G^{|V_k \cap V_m|} \leftarrow V_k \cap V_m$ 
13:         $G^*(s_i) += G^{|V_k \cap V_m|}$ 
14:      Fin Si
15:    Fin Para
16:  Fin Si
17: Fin Para

```

Los dos niveles:

- Primer nivel (Árbol de notificaciones/NT): Se debe de tener en cuenta que cada vértice es un grupo de nodos, según la Definición 2. La tarea principal de *NT* es evitar la redundancia y duplicación de datos. Ver algoritmo 2.
- Segundo nivel (Árbol de expansión jerárquica/HST): El mecanismo de informe de datos y el mecanismo de prolongación de vida útil son los dos objetivos principales de este nivel. El primer nivel logra el proceso de recopilación de datos.

Algoritmo 2 Construyendo el árbol de notificaciones.

Entrada: s_j

Salida: NT (s_j)

```

1:  $Root \leftarrow s_j$ ;
2:  $G_j \leftarrow (V_j, E_j) // V_j = \{s_x \mid s_x \in G^*(s_j)\}$  y  $E_j \subseteq [V_j]^2$ 
3:  $G_{j,i} \leftarrow G_j - s_j // \{G_{j,0}, G_{j,1}, \dots, G_{j,i}\}$  subgráfos.
4: Para (int  $i \leftarrow 0$ ;  $i < |G_{j,i}|$ ;  $i++$ ) Hacer
5:    $T_{j,i} \leftarrow BuildSpanningTree(G_{j,i}) // construye el árbol de expansión para cada subgráfo$ 
6:    $Root.children.add(T_{j,i})$ 
7: Fin Para

```

Proceso de construcción de hst: El proceso de construcción de *HST* es basado en *NT*. Por conveniencia, sea $HST = \{L_0, L_1, L_2, \dots, L_x\}$ los niveles del

árbol, $1 \leq x \leq n/2$, donde $n = |S|$ es el número de nodos en la red. Cada nivel contiene un conjunto de nodos tal que $L_i = \{s_0, s_1, \dots\}$. Se supone que el nodo raíz se encuentra en el primer nivel del árbol. El proceso de construcción del *HST* se extiende desde el nodo raíz hasta las hojas (de arriba hacia abajo) de la siguiente manera:

1. Para el nodo raíz, se agrega a $L_0 = \{s_r\}$, y luego para cada nodo en el árbol de notificaciones $NT(s_r) = \{s_r \rightarrow s_{r+0} \rightarrow s_{r+1} \rightarrow \dots\}$; si s_{r+i} no es marcado, entonces se agrega s_{r+i} a L_1 y s_{r+i} será un hijo de s_r .
2. Para otros nodos en otros niveles el método de procesamiento es el mismo que el del primer nivel. La única diferencia es que el primer nivel contiene un solo nodo, es decir, la raíz.

Mecanismo de reporte de datos usando hst: El árbol de notificaciones agrega los datos y el nodo proxy comenzará el proceso de reenvío. Cualquier nodo en la ruta del proxy al nodo raíz reenvía los datos recopilados a su padre en *HST* o a su padre en *NT*. El padre en *HST* es el predeterminado. Si la energía del padre en *HST* es mayor que el valor del umbral, entonces se selecciona el padre en *HST*. De lo contrario, se selecciona el padre en *NT* (ver algoritmo 3).

Algoritmo 3 Reporte de datos.

Entrada: Nodo informador.

Salida: Los datos del nodo informador se envían a su padre.

```
1: DataReport(ReporterNode, pkg)
2: ParentInHST ← ReporterNode.ParentSensor;
3: Si (ParentInHST.BatteryPercentage > ParentInHST.Energythreshold) entonces
4:   ReporterNode.SendData(ParentInHST, pkg);
5:   DataReport(ParentInHST, pkg); // llamada recursiva
6: Si no // Seleccione un nodo del NT del nodo informador
7:   ParentInNT ← SwitchOperation(ReporterNode, ParentInHST); // ver algoritmo
   7
8:   ReporterNode.SendData(ParentInNT, pkg);
9:   ParentInNT.ReceiveData(ReporterNode, pkg);
10:  DataReport(ParentInNT, pkg) // llamada recursiva
11: Fin Si
```

Enrutamiento de datos: Enrutamiento de datos simple basado en el modelo de agrupación. Cuando un nodo fuente tiene datos que enviar, primero comienza buscando el grupo de distancia mínima a la estación base (algoritmo 4) y luego desde el grupo seleccionado elige el siguiente salto para reenviar los datos. El siguiente salto se puede seleccionar de acuerdo con la siguiente distancia mínima al siguiente nodo objetivo o la energía máxima del nodo objetivo. Después de mantener el siguiente salto, el líder del grupo seleccionado transmitirá los datos directamente al nodo objetivo.

Algoritmo 4 Enrutamiento de datos.**Entrada:** s_i es el nodo fuente**Salida:** envía datos a EB directamente o reenvía datos a través de una ruta de múltiples saltos.

```

1: DataRouting( $s_i$ , Message msg)
2: TargetGroup  $\leftarrow$  GetMinDistanceGroupFor( $s_i$ );
3:  $s_t \leftarrow$  MinDistanceSensor(TargetGroup); //  $s_t$  es el nodo destino.
4: Si ( $s_t.ID \neq$  SinkNode.ID) entonces
5:    $s_i.SendData(s_t)$ ;
6:    $s_t.ReceiveData(s_i)$ ;
7:   DataRouting( $s_t$ , msg);
8: Si no
9:    $s_i.SendData$ (SinkNode);
10: SinkNode.ReceiveData( $s_i$ );
11: Fin Si

```

Tabla de objetos detectados (SOT por sus siglas en inglés) y tabla de mensajes de notificación (NMT por sus siglas en inglés) : Para $s_j \in S$ se asume que tiene una Tabla de Objetos Detectados $SOT(s_j)$ para guardar la información del objeto detectado (es decir, posición actual), y tiene una Tabla de Mensajes de Notificación $NMT(s_j)$ para almacenar la notificación enviada/recibida-hacia/desde otros nodos. Cuando un objeto $o_i \in M$ es detectado por s_j , s_j puede crear una nueva entrada en $SOT(s_j)$ para o_i si y solo si s_j no tiene ningún registro de notificación recibido de otro $s_x \in G^*(s_j)$. Cuando s_j crea una nueva entrada en $SOT(s_j)$ enviará un mensaje de notificación de insertar a $G^*(s_j)$ a través del árbol de notificaciones $NT(s_j)$ para notificar a todos los nodos en $G^*(s_j)$ y pedirles que agreguen este mensaje de notificación a sus tablas NMT; por lo tanto los otros nodos en $G^*(s_j)$ no detectarán o_i hasta que s_j envíe un mensaje de notificación de eliminación que indique que o_i salió de su rango.

En este momento, todos los nodos en $G^*(s_j)$ eliminarán los mensajes de notificación de o_i y entrarán en cualquier otro nodo en $G^*(s_j)$. La selección del nodo proxy se puede determinar en función de los registros de NMT; por ejemplo, cuando hay más de un registro para o_i en $NMT(s_j) = \{N_{i,j}, N_{i,0}, N_{i,1}, \dots, N_{i,x}\}$ $x \neq j$, $N_{i,x}$ es una notificación de mensajes para el objeto o_i que es detectado por el nodo s_x . En este caso s_j será el nodo proxy si cumple los requisitos, es decir, ID máximo, energía residual, el estado actual, etc., si s_j pierde esta ronda o no fue elegido como nodo proxy, eliminara el registro de $SOT(s_j)$ y elimina el mensaje de notificación $N_{i,j}$ de $NMT(s_j)$. Algo que se debe de destacar es que todos los grupos $G^*(s_j)$ tendrán la misma NMT (Tabla de mensajes de notificación).

4. Desarrollo de trabajo experimental

Para la evaluación del rendimiento, se ha desarrollado un simulador en Java. El desarrollo en Java se decidió ya que es un lenguaje multiplataforma y hay

innumerables bibliotecas y/o utilidades creadas para extender su funcionalidad, además de la experiencia previa con este lenguaje. El entorno de simulación cuenta con un área de trabajo, un panel de control, un apartado para el indicador de los estados de los nodos sensores y en la parte superior un menú para crear la configuración de la simulación, como se observa en la figura 1.

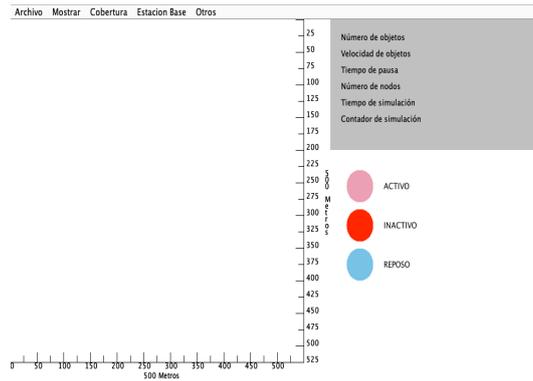


Fig. 1. Interfaz gráfica del simulador.

Las pruebas se desarrollaron con una base de datos sintética (contiene las tablas de sensores y topologías). La tabla sensores tiene los atributos de id, idTopologías, nodoID, pox, poy, poz y r (radio). La tabla de topologías tiene como atributos idTopologías y nombre (nombre de la topología).

Tabla 2. Parámetros de simulación (E_{elec} =Disipación de energía, ε_{mp} =Modelo multitrayecto del amplificador transmisor, EB=Estación base).

Parámetro	Valor
Tamaño de la red	450 x 450 m^2
Nodos	600 nodos
Radio	12 metros
Tamaño de datos	256 bits
Energía inicial	0.5 julios
E_{elec}	50 nJ/bit
ε_{mp}	0.001 pJ/bit/ m^2
Localización de EB	Estática

Cobertura de Topología: Cada sensor puede recopilar datos mediante el monitoreo de un área generalmente pequeña que se encuentra en su rango de detección. Decimos que el sensor proporciona *cobertura* a esta área. Uno de los problemas fundamentales de las redes de sensores inalámbricas es la cobertura

de los objetos en conjunción con las limitaciones de eficiencia de energía [8]. El problema de cobertura se describe como un problema de calidad de servicio, donde el objetivo es encontrar qué tan bien, en términos de calidad de datos de monitoreo, el campo es monitoreado por los sensores. Por otro lado, el problema se formula como la maximización de la vida útil de la red bajo el área o la restricción de cobertura. En la primera formulación (ver figura 2 a la izquierda), los sensores deben de monitorear toda el área (por ejemplo, una gran región cuadrada), mientras que en la segunda los sensores deben cubrir un conjunto de puntos (objetivos) que yacen en el campo (ver figura 2 de la derecha).

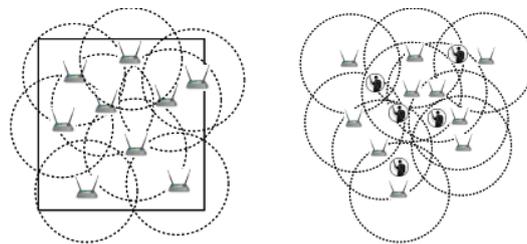


Fig. 2. Dos tipos principales de cobertura: cobertura de área y objetivo (el cuadro grande denota el área cubierta, los soldados a la derecha denotan los objetivos cubiertos).

Cobertura basada en cuadrícula: Para los esquemas de cobertura basados en cuadrática, se seleccionaron dos algoritmos proporcionados en [8], Versión 1 de cobertura cuadrática y la versión 2 de cobertura cuadrática.

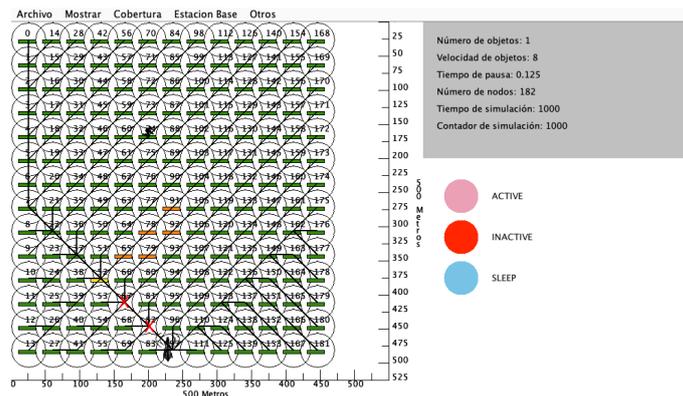


Fig. 3. Árbol de expansión jerárquica de 182 nodos desplegados utilizando el esquema de cobertura cuadrática 1. El rango de detección es de 25 metros y el rango de comunicación es de 50 metros. La barra de proceso indica la capacidad de la batería. El nodo de estación base es 97.

Cobertura basada en Zigzag: En el esquema de cobertura zigzag [5], el área de interés se divide en múltiples patrones de zigzag con múltiples esquinas y segmentos de líneas, cada nodo se despliega en una esquina del patrón de zigzag. Ver figura 4.

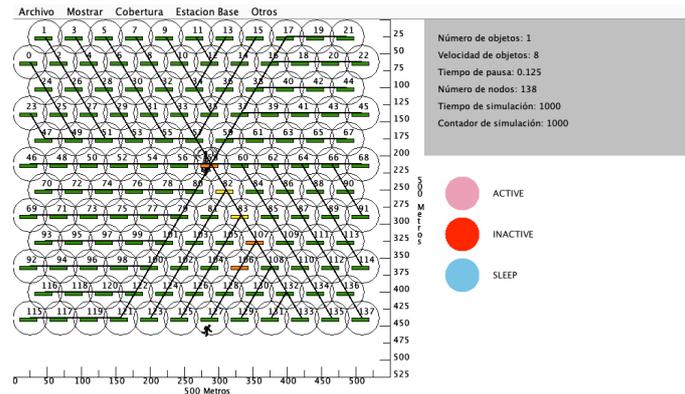


Fig. 4. Árbol de expansión jerárquica de 138 nodos desplegados utilizando el esquema en zigzag, el rango de detección es de 25 metros, mientras que el rango de comunicación es de 50 metros. La barra de proceso indica la capacidad de la batería. El nodo de estación base es 58.

Consumo de Energía: Se utiliza un modelo simple llamado "Modelo de radio de primer orden" [9]. En GLT, *HST* consume la mayor parte de la energía durante el enrutamiento de datos. *NT* consume una menor cantidad de energía.

4.1. Comparación de resultados obtenidos

Para la evaluación, se realizó un análisis cuantitativo. Esta evaluación se llevó a cabo con los parámetros descritos en la tabla 2. Además, el análisis evalúa la eficiencia haciendo énfasis a cuatro rubros en general; 1) número paquetes transmitidos, 2) consumo promedio de energía de la recolección de datos versus el número de nodos, 3) vida útil de la red y 4) consumo de energía con respecto al tiempo. La evaluación se realizó haciendo comparación con los protocolos de agrupación en clústeres existentes (CAICS por sus siglas en inglés [2]), (DLC por sus siglas en inglés [11]) y (MHC por sus siglas en inglés [14]) de acuerdo en la literatura. Los parámetros predeterminados utilizados en esta sección son los mismos que [2]. La estación base está ubicada en el centro del campo (200, 200). Las unidades de medida son el tiempo de simulación (segundos), el retardo de transmisión (milisegundos), el consumo de energía (julios) y la velocidad del objeto.

La eficiencia del envío de paquetes en el algoritmo GLT con respecto a CAIC, MHC y DLC se hace notorio, debido a que el algoritmo GLT ubica la estación

base en el centro de la detección. Cuando la estación base está ubicada en el centro del campo de detección, la vida útil de la red es más duradera y el número de paquetes de operación de conmutación se reduce.

En contraste, cuando la ubicación del receptor se encuentra a los extremos del campo de detección, la vida útil de la red se reduce considerablemente y como resultado, el número de paquetes de operación de conmutación aumentará para prolongar la vida útil de la red el mayor tiempo posible.

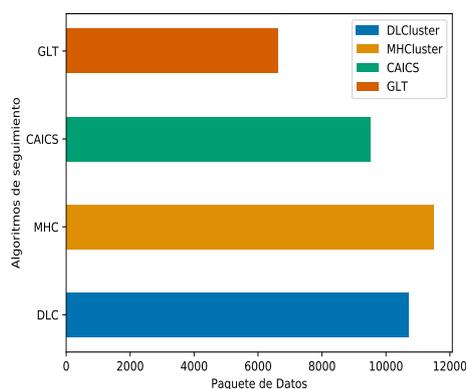


Fig. 5. Número de paquetes de datos transmitidos.

El tamaño de la red (es decir, el número de nodos en la red) tiene un impacto en el retraso de la transmisión y el consumo de energía en los cuatro enfoques (es decir, GLT, CAICS, DLC y MHC). En el caso de GLT, cuanto mayor sea el tamaño de la red, mayor será el número de grupos.

Por lo tanto, el tamaño de NT y HST será mayor. Por otro lado, en las redes basadas en clúster, el número de miembros del clúster aumentara en cada grupo, lo que conduce a más nodos involucrados en la detección y el seguimiento. El consumo medio de energía y el retraso medio de transmisión de la recopilación de datos frente al número de nodos se muestran en la figura 6.

5. Conclusiones

Las empresas en México optan por nuevas tecnologías para optimizar e innovar sus procesos y estar a la altura de los mercados internacionales. Las redes de sensores es una de las aplicaciones clave que está ampliamente adaptada debido a su gran cantidad de implementaciones y usos. Dado que el principal obstáculo que enfrenta la mayoría de las técnicas de seguimiento en redes de sensores es el hecho de que la red de sensores sufre de una fuente de alimentación muy limitada que a su vez restringe el tipo de tareas que debe realizar la red de sensores. El objetivo principal de este trabajo se cumple, ya que la durabilidad de la red y la conservación de energía son atacados con la implementación de

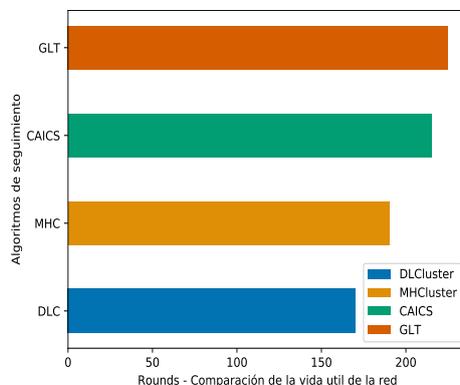


Fig. 6. Consumo promedio de energía de la recolección de datos versus el número de nodo.

los algoritmos planteados, siendo así que este trabajo se puede tomar como referencia para poner en marcha en ambientes productivos manufactureros.

Se crearon topologías para tener el enfoque de redes de sensores, los resultados de la simulación reflejan una disminución de sobrecarga de comunicación sin disminuir la precisión del seguimiento del objeto y se logra una buena extensión de la vida útil de la red. Los nodos en el árbol de notificaciones se activarán para garantizar la precisión del seguimiento, pero sus datos recopilados se transmitirán al nodo proxy en un solo salto. Para trabajos futuros se propone el estudio del movimiento rápido de los objetos.

Referencias

1. Chen, T.S., Chen, J.J., Wu, C.H.: Distributed object tracking using moving trajectories in wireless sensor networks. *Wireless Networks* 22(7), 2415–2437 (Oct 2016)
2. Chen, W.P., Hou, J., Sha, L.: Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing* 3(3), 258–271 (Jul 2004)
3. Faudó, I.E., Cota-Ruiz, J., Aguilera, F.J.E., Perea, P.R.: Análisis y comparación de algoritmos de localización en redes inalámbricas de sensores. *Pistas Educativas, Instituto Tecnológico de Celaya* 108 (2014)
4. Feng, J., Lian, B., Zhao, H.: Coordinated and Adaptive Information Collecting in Target Tracking Wireless Sensor Networks. *IEEE Sensors Journal* 15(6), 3436–3445 (Jun 2015)
5. Hawbani, A., Wang, X.: Zigzag Coverage Scheme Algorithm & Analysis for Wireless Sensor Networks. *Network Protocols and Algorithms* p. 19 (Dec 2013), <http://www.macrothink.org/journal/index.php/npa/article/view/4688>
6. Hawbani, A., Wang, X., Karmoshi, S., Kuhlani, H., Ghannami, A., Abudukelimu, A., Ghoul, R.: GLT: Grouping Based Location Tracking for Object Tracking Sensor Networks. *Wireless Communications and Mobile Computing* (2017)

7. Hawbani, A., Wang, X., Karmoshi, S., Wang, L., Husaini, N.: Sensors Grouping Hierarchy Structure (GHS) for Wireless Sensor Network. *International Journal of Distributed Sensor Networks* 11(8), 650519 (2015)
8. Hawbani, A., Wang, X., Naji, A., Karmoshi, S.: Grid Coverage Algorithm & Analyzing for wireless sensor networks. *Network Protocols And Algorithms* 3, 1 (Mar 2014)
9. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. pp. 10 pp. vol.2- (Jan 2000)
10. Liao, W., Chang, K., Kedia, S.P.: An object tracking scheme for wireless sensor networks using data mining mechanism. In: *2012 IEEE Network Operations and Management Symposium*. pp. 526–529 (April 2012)
11. Mamun, Q., Ramakrishnan, S., Srinivasan, B.: Selecting Member Nodes in a Chain Oriented WSN. In: *2010 IEEE Wireless Communication and Networking Conference*. pp. 1–6 (Apr 2010)
12. Raga, M.C.M.: *Filtro de Kalman y sus aplicaciones* (2018)
13. Weiser, M.: *The computer for the 21 st century*. *Scientific American* (1991)
14. Xue, W., Jun-Jie, M., Sheng, W., Dao-Wei, B.: Cluster-based Dynamic Energy Management for Collaborative Target Tracking in Wireless Sensor Networks. *Sensors* 7 (Jul 2007)
15. Zheng, Y., Zhou, X. (eds.): *Computing with spatial trajectories*. Springer, New York (2011)