

## Selección de un Gateway de respaldo, adaptando el algoritmo Omega en Redes 6LowPAN

Cristian Fabricio Espinosa Gualotuña<sup>a</sup>, Carlos Roberto Egas Acosta<sup>b</sup>

<sup>a</sup> Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Ladrón de Guevara E11-253, Quito, Ecuador  
crisfabr26@hotmail.com

<sup>b</sup> Departamento de Telecomunicaciones y Redes de la Información, Escuela Politécnica Nacional, Ladrón de Guevara E11-253, Quito, Ecuador  
cegas@ieee.org

**Resumen.** Teniendo en cuenta los nuevos desafíos a los que las redes actuales se enfrentan y considerando un alto incremento en la implementación de redes de sensores inalámbricos, se propone mejorar la confiabilidad del acceso a internet de una red sensor inalámbrica, adaptando el algoritmo Omega a este tipo de ambientes; y así, comprobar su funcionamiento con la implementación de un prototipo de red 6LowPAN. Para lo cual se utilizará el hardware desarrollado por Libelium y la plataforma de desarrollo de IBM, denominada Mote Runner para la configuración, depuración y verificación de la solución propuesta.

Los resultados fueron obtenidos implementando el algoritmo propuesto en cada nodo sensor inalámbrico, tomando como parámetros el nivel de batería de los nodos y su identificador, lo que permitió la elección de un Gateway de respaldo de entre los posibles nodos candidatos. En el caso de que el nodo Gateway seleccionado en un inicio del funcionamiento de la red, se encuentre indisponible, la utilización del algoritmo propuesto permite tener una alternativa de selección de un nodo Gateway de respaldo, con lo cual se pretende, brindar mayor confiabilidad en la conectividad de este tipo de redes con el internet.

**Palabras Clave:** Algoritmo, 6LowPAN, Redes de Sensores Inalámbricos, Quality of Service

### 1 Introducción

El desarrollo acelerado de las redes de datos, las tecnologías de acceso inalámbrico y de última milla, tales como: IEEE 802.11, IEEE 802.15.4 y los servicios ofrecidos por operadores de telefonía móvil, permiten que cualquier usuario pueda acceder a los datos de una red de sensores inalámbricos dando lugar a una infinidad de aplicaciones que están dentro del “Internet de las cosas – IoT (Internet of Things)”.

Las exigencias actuales desafían a casi todos los dispositivos tradicionales; sin embargo, el uso de aplicaciones inalámbricas y el desarrollo acelerado de las redes inalámbricas de alto rendimiento, está permitiendo sobrellevar las limitaciones en los dispositivos actualmente utilizados; como lo es, la facilidad de direccionamiento que proporciona el protocolo de red IPv6. En ambientes industriales se ha llevado a cabo

al desarrollo de las denominadas Redes de Sensores Inalámbricas Industriales, que en este momento son el centro de atención de muchos grupos de investigación. En este contexto, las redes son importantes en la infraestructura de las empresas, hogares e instituciones educativas, por lo que es necesario trabajar en el desarrollo de redes de sensores inalámbricos, que permitan cubrir la calidad de servicio que hoy en día las redes cableadas brindan a los usuarios.

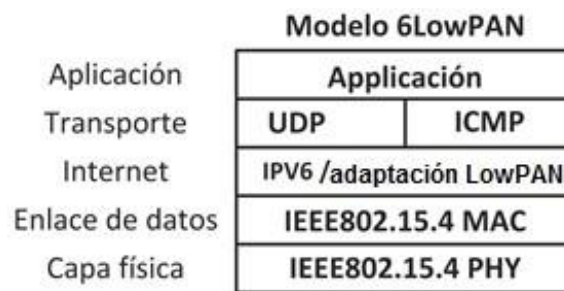
Por lo expuesto, desde hace algún tiempo atrás empresas dedicadas al Networking, han venido trabajando en el desarrollo de redes IPv6, a fin de que ayuden al cumplimiento de calidad de servicio y además sean altamente rentables, fomentando así, el desarrollo de aplicaciones inalámbricas bajo IPv6 con todas las ventajas que este protocolo supone.

Gracias a los esfuerzos de la empresa International Business Machines Corp (IBM) en conjunto con la empresa Libelium, se ha desarrollado el hardware para trabajar en redes 6LoWPAN denominado “Kit de desarrollo Waspote PRO V1.2”, facilitando el emprendimiento en el desarrollo de redes de sensores inalámbricos que utilicen el protocolo de red IPv6; permitiendo el desarrollo de soluciones menos costosas.

## 2 Ambiente de desarrollo

### 2.1 Estándar 6LoWPAN

6LoWPAN (IPv6 Over Low Power Wireless Personal Area Networks) es un estándar que posibilita el uso de IPv6 sobre redes basadas en el estándar IEEE 802.15.4. Este protocolo permite que dispositivos conectados al internet puedan tener conectividad con los nodos de una red de sensores inalámbrica de una manera directa. Los nodos de la red sensor inalámbrica son dispositivos de baja potencia, que forman parte de las redes de área personal (Personal Area Networks PAN) [1]; y que al utilizar IPV6 como protocolo de red, se denominan redes 6LoWPAN. En la Fig. 1 se muestra la arquitectura de la red 6LoWPAN [2].



**Fig. 1.** Arquitectura 6LoWPAN.

El reto inicial de 6LoWPAN fue que un encabezado IPv6 contenga 40 octetos, a comparación de la máxima longitud de una trama en el estándar IEEE 802.15.4 el cual

contiene 127 octetos; permitiendo el uso de encriptación, el espacio de datos a transferir se reduce tanto, lo cual ocasiona un encabezado de tamaño más grande que la carga útil, debido a los bits adicionales resultado de la encriptación [2] [3]. Si se empleara la carga útil para transportar el encabezado de IPv6 quedarían solo 33 octetos para los datos encriptados, lo cual es sumamente ineficiente, en este contexto y a fin de lograr una eficiente transferencia de datos, el encabezado de 40 octetos de IPv6 se comprime para que ocupe el menor espacio posible de la carga útil de una trama IEEE802.15.4 [4]. Por otra parte, considerando la corta longitud de la carga útil, otra dificultad principal es que IPv6 contempla MTU's (Maximun Transfer Units) de 1280 octetos que implica la fragmentación de paquetes IP en tramas que utilicen el estándar IEEE 802.15.4. En cualquier caso, 6LowPAN incluye un campo codificado de control llamado "dispatch" establecido en la RFC 4944 con bits que indican la coexistencia con otras redes, el tipo de compresión del encabezado y secuencias de la fragmentación [5].

## 2.2 Algoritmo Omega

Chandra propuso el detector de fallos Omega  $\Omega$ . Este detector de fallos garantiza que eventualmente todos los procesos correctos siempre tienen el mismo proceso correcto como líder [6] [7] [8]. Chandra y Toueg definen el detector de fallos ( $\diamond P$ ), como eventualmente perfecto. Este detector de fallos garantiza que hay un tiempo después de lo cual cada proceso sólo confía en los procesos correctos [7]. En él se demuestra que si los procesos no conocen a priori todos los miembros (procesos) del sistema, es decir, el número de miembros es desconocido, es imposible sospechar finalmente todos los procesos defectuosos. Por ejemplo, si se conoce el número de miembros y los procesos  $p_i$  dañados en el inicio de un escenario, es decir antes de enviar o recibir cualquier mensaje, esto imposibilita para que otros procesos  $p_j$  puedan aprender acerca de los procesos  $p_i$  en el sistema, y por lo tanto, es imposible para los proceso  $p_j$  ver a los procesos  $p_i$  sospechosos [6] [7].

Sin embargo, se puede tener un sistema  $S$  denotado como  $\diamond P$  de tal manera que hay un tiempo después de lo cual cada proceso correcto permanentemente sólo confía en todos los procesos correctos [7]. El funcionamiento del algoritmo Omega se resume en la Figura 2.

<b>initialization:</b>	<b>task T2:</b>
(01) $correct_i \leftarrow \{i\};$	(13) <b>upon reception of message</b> ( $set_j$ ) : $j \neq i$ <b>do</b>
(02) $membership_i \leftarrow \{i\};$	(14) <b>for_each</b> ( $k \in set_j$ : $k \neq i$ ) <b>do</b>
(03) $leader_i \leftarrow i;$	(15) <b>if</b> ( $k \notin membership_i$ ) <b>then</b>
(04) <b>start</b> tasks T1 and T2;	(16) $membership_i \leftarrow membership_i \cup \{k\};$
<b>task T1:</b>	(17) <b>create</b> $timer_i(k)$ and $timeout_i[k];$
(05) <b>repeat forever</b> each $\eta$ time	(18) $timeout_i[k] \leftarrow 1;$
(06) <b>if</b> ( $ correct_i  > n/2$ ) <b>then</b>	(19) <b>end_if</b>
(07) $successor_i \leftarrow next\_to\_i\_in\_correct_i();$	(20) $correct_i \leftarrow correct_i \cup \{k\};$
(08) <b>send</b> ( $correct_i$ ) to $successor_i$ ;	(21) <b>set</b> $timer_i(k)$ to $timeout_i[k];$
(09) <b>else</b>	(22) <b>end_for</b>
(10) <b>broadcast</b> ( $\{i\}$ );	(23) $leader_i \leftarrow \min(correct_i);$
(11) <b>end_if</b>	(24) <b>upon expiration of</b> $timer_i(k)$ :
(12) <b>end_repeat</b>	(25) $timeout_i[k] \leftarrow timeout_i[k] + 1;$
	(26) $correct_i \leftarrow correct_i \setminus \{k\};$
	(27) $leader_i \leftarrow \min(correct_i);$

Fig. 2 Algoritmo Omega

En la Fig. 2, cada proceso  $p_i$  inicialmente envía broadcast “latidos” varias veces para indicar que el proceso está vivo (línea 10). Para conocer qué procesos están en el sistema, los procesos  $p_i$  tienen un conjunto de miembros, que inicialmente solo contiene a sí mismo. El conjunto  $Correct_i$  mantiene los procesos de los que considera que están vivos, siempre que contenga, al menos a sí mismo. Si la mayoría de los procesos están vivos, entonces eventualmente  $|correct_i| > n/2$  y el proceso  $p_i$  enviará latidos periódicamente al sucesor (línea 08). La variable  $sucesor_i$  contiene el proceso que devuelve la función  $next\_to\_i\_in\_correct_i()$  (línea 07) [6] [7].

La función antes mencionada obtiene el identificador del proceso más cercano al identificador de  $p_i$  en la secuencia formada por todos los elementos  $correct_i$ , en orden creciente y cíclico. Por ejemplo, si el proceso  $p_3$  tiene  $correct_3 = \{1, 3, 5, 8\}$ , entonces  $next\_to\_i\_in\_correct_3()$  devolverá 5. En otro ejemplo, si el proceso  $p_3 = \{1, 2, 3\}$ , entonces  $next\_to\_i\_in\_correct_3()$  lo hará volver 1. Como se muestra, si  $|correct_i| > n/2$ , cada  $p_i$  proceso correcto eventualmente enviará mensajes sólo a un proceso correcto, estableciendo un ciclo formado por todos los procesos correctos. La variable  $i$  tiene el identificador del proceso que  $p_i$  considera su líder. Su valor es el valor más pequeño de  $correct_i$  [6] [7]. Esto se lo realiza en la Tarea 1 (Task 1) [7] [9].

En la tarea 2 (Task 2), se tiene el proceso de recepción y traspaso de los procesos a través de mensajes, en este momento se van agregando a  $membership_i$  los procesos correctos que aún no han sido agregados, para este propósito en la línea 15 se ejecuta el lazo  $for\_each()$ , con el fin de poder ir agregando todos los procesos correctos; para conseguir esta meta también se hace el uso de temporizadores  $timer$  con su debido  $timeout$ , dichos temporizadores permitirán verificar los procesos correctos, todo esto se lo ejecuta para que todos los procesos lleguen a conocer el proceso que se eligió como líder, cuyo valor es el más pequeño de  $correct_i$ , el mismo que se almacenará en  $leader_i$  [6] [7] [10].

### 3 Formulación del Problema

La red de sensores inalámbricos 6LowPAN se conforma de los nodos inalámbricos y el nodo Gateway, el mismo se conecta a Internet u a otra red remota. En el caso de que el nodo Gateway falle, la red sensor inalámbrica no tendría conectividad con Internet, por lo que es necesario tener varios Gateways de respaldo, de los cuales, uno de ellos entre a funcionar cuando el principal falle. Existen algunos mecanismos que podrían ser utilizados para seleccionar al Gateway de respaldo, en el presente trabajo se ha adaptado el algoritmo Omega para seleccionar uno de los Gateways, como un nodo Gateway líder de respaldo, con el propósito de que el nodo seleccionado pueda ser el encargado de enviar los datos si existe algún fallo del nodo Gateway inicial y de esta manera tener un Gateway de respaldo, para que no exista una pérdida considerable de datos, y aumentos de retardos por pérdida de la conectividad de la red. La ejecución del algoritmo Omega permitirá un aumento en la confiabilidad de la conectividad de la red 6LowPAN con el internet, ya que por defecto tienen un solo nodo Gateway para su funcionamiento.

Los parámetros a tomar en cuenta para la selección del nodo líder de respaldo son: el estado de batería más alto de todos los nodos que conforman el prototipo de red de

sensores inalámbricos 6LowPAN y que pueden actuar como Gateways, en el caso de que más de un nodo tenga el mismo estado de batería, la aplicación desarrollada va a discriminar mediante el identificador único de cada nodo, cuyo valor será el más alto, permitiendo la selección del nodo líder de respaldo, la finalización de la ejecución del algoritmo Omega entrega como resultado el nodo seleccionado como líder, mismo que será el que tenga el estado de batería más alto y el identificador único más alto dentro de todos los nodos que conforman el prototipo de red 6LowPAN.

#### **4 Limitaciones para la implementación del algoritmo Omega en redes 6LowPAN.**

Mediante el análisis previo del Algoritmo Omega, se llegan a algunas consideraciones muy importantes a tomarse en cuenta, a fin de que éste pueda ser adaptado en un ambiente de red 6LowPAN. Las consideraciones se detallan a continuación:

1. El algoritmo Omega está orientado a procesos ejecutados a nivel de aplicación, para detectar fallos y elegir un proceso líder, en el caso de este proyecto el objetivo es adaptar dicho algoritmo al prototipo de red de sensores inalámbricos 6LowPAN y reemplazar los procesos por nodos que conforman el prototipo de red. Los nodos deben ejecutar el algoritmo Omega modificado para la selección de un nodo líder de respaldo.

2. El protocolo de red IPv6 no trabaja con direcciones Broadcast, ya que dispone para su uso únicamente direcciones Multicast, Unicast, Anycast. Por este motivo se diseña un “broadcast propietario”, que enviará datos que identifiquen a cada nodo inalámbrico, permita verificar si cada nodo está activo, y envíe datos de estado de la batería e identificador del nodo, que permitirán cumplir con parte de la Tarea 2 (Task 2) del algoritmo Omega, la cual en el algoritmo original es otro hilo de programación diferente.

Al no poder enviar un broadcast para poder monitorizar que cada nodo inalámbrico esté “vivo” como se hace en el algoritmo Omega original, se diseña la aplicación dentro del software desarrollado, para ahorrar procesamiento y consumo de batería en los nodos inalámbricos. Dicho “Broadcast propietario” implicará el envío de toda la información requerida entre los nodos en una conexión punto a punto, y reducir el consumo de batería. La modificación realizada en la aplicación consistirá que al mismo tiempo que podrá conocer si los nodos inalámbricos están “vivos”, también permitirá la recolección de datos del estado de batería de cada nodo y su identificador único dentro del prototipo de red, y de esta manera adaptar el algoritmo a los requerimientos de la red 6LowPan para la elección de un nodo líder de respaldo.

3. Los tiempos para los procesos: El algoritmo Omega es un algoritmo orientado a procesos, el mismo toma en cuenta los tiempos de cada proceso para cumplir su objetivo que es la elección de un proceso líder y que los mensajes puedan llegar a tiempo en este caso, estos tiempos no se pueden controlar de forma exacta, ya que el proceso depende del hardware del nodo, esta es una limitante que influenciará en la aplicación desarrollada, así como limitantes de memoria, tiempos de procesamiento, capacidad de procesar las líneas de código. Los nodos que conforman el prototipo de red 6LowPAN, deben ejecutar individualmente la aplicación desarrollada.

4. Protocolo IPv6: Al trabajar con este protocolo, se restringe ciertos parámetros del algoritmo Omega original, ya que tiene campos específicos y tipos de direcciones que van a limitar a la aplicación desarrollada para el prototipo de red 6LowPAN, como es el caso de que se desee trabajar con direcciones de 128 bits, lo que significa más procesamiento. Para minimizar la afectación y compensar el direccionamiento del protocolo de red IPv6 de 128 bits, se va a utilizar el encabezado comprimido que especifica el estándar 6LowPAN al momento de la adaptación del algoritmo Omega, la información que genere cada nodo, puede ser enviada a una red IPv6 remota o a un servidor remoto donde los datos obtenidos sean recolectados y procesados.

5. Restricciones en la programación: Debido a los limitados recursos en plataformas integradas, Mote Runner no es capaz de soportar todas las características del lenguaje C# que son utilizadas en la implementación del algoritmo original. Por lo tanto nos vemos obligados a utilizar la definición de un subconjunto de los lenguajes Java o C#, con los cuales trabaja la plataforma de desarrollo Moterunner.

A continuación, se muestra las restricciones más prominentes y omisiones al momento de escribir aplicaciones para Mote Runner en el lenguaje que fue desarrollado la aplicación para el presente proyecto C# (C Sharp):

- ✓ Aritmética de enteros de 64 bits o más.
- ✓ Matrices multidimensionales y Reflexión.
- ✓ Las clases internas en C # sólo se admiten las clases internas en Java.
- ✓ Limitadas Templates .
- ✓ Delegados de multidifusión o multicast, solo es compatible el delegado unicast.
- ✓ Hilos y sincronización de primitivas.
- ✓ Tipo de datos bool, String y String Array double y float ✓ Empaquetamiento.
- ✓ API's de tiempo de ejecución estándar y Enumeraciones.

6. Limitantes de Hardware: Estas limitantes son principalmente por la cantidad de memoria RAM que contiene cada nodo, lo que conlleva a tiempos considerables en aplicaciones, como es nuestro caso ya que involucra procesos de comunicación de los nodos, el proceso de "Broadcast propietario", la ejecución del algoritmo adaptado, la mismo que debe lograr un ordenamiento de los datos para cumplir con el objetivo fundamental que consiste en la elección de un nodo Gateway de respaldo, las capacidades de almacenamiento es otro aspecto a tener en cuenta, razón por la cual hay que implementar aplicaciones óptimas, hablando en cuestiones de programación.

## 5 Adaptación del algoritmo Omega en ambientes 6LowPAN

Adaptación del algoritmo Omega	
<b>Inicialización</b>	
a.	estadoBateria[n]
b.	miembros[n]
c.	LIDER, valor 1
d.	SUCESOR, valor 0
e.	bateriaSucesor, valor 0
f.	TIEMPO_BROADCAST, valor 1
g.	MAX_MOTES, valor 0
<b>Algoritmo</b>	
h.	<b>for</b> (m = 0; m < MAX_MOTES; m++)
i.	<b>for</b> (n = 0; n < m; n++)
j.	Intercambio de valores, estadoBateria[]
k.	<b>end</b>
l.	<b>end</b>
m.	<b>if</b> (miembros[0]==LIDER) <b>then</b>
n.	SUCESOR = miembros[1]
o.	BATERIA_SUCESOR = estadoBateria[1]
p.	<b>Else</b>
q.	SUCESOR = miembros[0]
r.	BATERIA_SUCESOR = estadoBateria[0]
s.	<b>end-if</b>
<b>Broadcast</b>	

**Fig. 3.** Secuencia de procesos en la adaptación del algoritmo Omega adaptados para la red 6LowPAN

En la Fig. 3, se aprecia la adaptación del algoritmo propuesto, para lo cual, se inicializan los vectores y las variables como son el vector ESTADO DE BATERIA, el vector MIEMBROS, las variables LÍDER, SUCESOR, BATERIA SUCESOR, TIEMPO DE BROADCAST y NÚMERO MÁXIMO DE NODOS. La identificación de los nodos fallidos por causa agotamiento de batería o daño en el enlace se lo deja a 6LowPAN, con el fin de reutilizar procesos ya creados. El nodo en un inicio definido Gateway líder, almacenado en la variable LIDER, va a solicitar el número máximo de nodos existentes dentro del prototipo 6LowPAN y lo almacenará en el vector NÚMERO MÁXIMO DE NODOS, con esto cada nodo procede hacer un proceso de “broadcast propietario”, mismo que deberá ser implementado a través del desarrollo en la aplicación, este “broadcast” utiliza una conexión punto a punto con todos los nodos existentes en el prototipo de red 6LowPAN.

Con base a lo citado, cada nodo inalámbrico envía información de su estado de batería y de su identificador único, este proceso es realizado por cada uno de los nodos que conforman el prototipo, toda esta información se almacena en cada nodo en el vector MIEMBROS y ESTADO DE BATERIA. Para hacer una comparación todo este proceso condensaría la Tarea 1 (Task 1) del algoritmo Omega original y una parte de la sección del traspaso de mensajes, correspondientes a la Tarea 2 (Task 2).

Cuando todos los nodos inalámbricos hayan enviado su estado de batería y su identificador único a sus vecinos, se inicia el proceso de ordenamiento de cada uno de

los datos relevantes mediante el algoritmo de la burbuja conocido e implementado en varios lenguajes de programación, lo cual es aplicado a todos los nodos de la red. Cumplido lo antes mencionado y una vez ordenado los datos, se realiza la comparación del estado de batería del sucesor definido en un inicio con la batería del sucesor elegido, esto en vista de que si dos o más nodos contienen el mismo estado de batería, se elegirá al nodo líder de respaldo mediante el identificador único de cada nodo, cuyo valor sea el más alto; si bien el algoritmo Omega original escoge como proceso líder al que tenga el identificador más bajo, nosotros seleccionamos arbitrariamente el identificador único de cada nodo, cuyo valor sea el más alto.

Después de haber efectuado las debidas comparaciones del estado de batería, considerando que si dos o más nodos inalámbricos poseen el mismo valor en su estado de batería, el nodo elegido como nodo líder de respaldo se almacenará en el vector SUCESOR, así como el valor de su batería en el vector BATERIA SUCESOR, dejando encendido un led indicador del nodo seleccionado como sucesor o líder de respaldo y todos los demás nodos permanecerán apagados. Todo el proceso descrito condensaría la Tarea 2 (Task 2) del algoritmo Omega que en forma general compete a la recepción y traspaso de mensajes.

Una vez adaptado el algoritmo Omega para ser utilizado en el prototipo de red 6LowPAN, se pueden señalar los cambios con respecto al algoritmo original. Uno de ellos y muy importante es la implementación del “Broadcast propietario”, que a su vez con la implementación del algoritmo de la burbuja, el cual no es contemplado en el algoritmo original pero indispensable en nuestra propuesta permitiendo, actuar en forma conjunta, para un ordenamiento de todos los nodos en base a su estado de batería y su identificador único, y posterior traspaso de mensajes entre cada nodo que conforma la red y de esta manera seleccionar el nodo líder de respaldo o nodo Gateway de respaldo.

Otro importante cambio con respecto al algoritmo Omega original el cual esta orientado a procesos; nosotros lo hemos adaptado para que este orientado nodos en una red 6LowPAN, además en el algoritmo original, el uso de tiempos y timers son de manera exacta, ésta es otra limitante dentro de la adaptación del algoritmo Omega al prototipo de red antes mencionado, ya que los tiempos estarán condicionados especialmente a las limitantes del hardware de cada uno de los nodos.

## **6 Adaptación del algoritmo Omega en el prototipo de red 6LowPAN**

La topología del prototipo en el cual se realizaron las pruebas se presenta en la Fig. 4, en dicha topología se asume que todos los nodos tienen conexión al Internet, el proceso mediante el cual se conmuta el enrutamiento del nodo Gateway caído al nuevo nodo Gateway no se considera en este estudio, ya que el mismo puede ser implementado con protocolos de enrutamiento. Además, es importante anotar que la topología de comunicación del algoritmo Omega es en malla, y las condiciones en el prototipo de red implican una topología tipo árbol o estrella

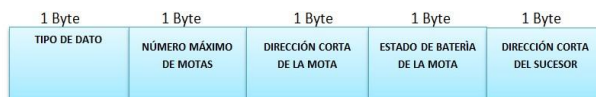




**Fig. 4.** Prototipo de red 6LoWPAN a implementarse

El escenario que se presenta en la Fig. 4, consistirá en 4 nodos inalámbricos que pueden ser utilizados como nodos Gateways, El requerimiento principal es que los nodos inalámbricos puedan comunicarse con el nodo Gateway que inicialmente está activo. La PC es utilizada para la configuración y gestión de los nodos. El nodo Gateway en el prototipo se conforma de la parte inalámbrica que le permite realizar la red 6LoWPAN, así como de la parte cableada que permitirá la conexión del Gateway con la PC, utilizando un túnel de IPv6/IPv4., debido a que la interfaz LAN del nodo Gateway del kit de desarrollo Waspnote PRO v1.2 se configura únicamente con IPv4, razón por la cual, la conexión del nodo Gateway con la PC debe ser realizado mediante dicho esquema de direccionamiento.

Para la ejecución de los procesos del algoritmo adaptado en los nodos sensores inalámbricos fue necesario crear una trama como se muestra en la Fig. 5, la misma que permita la transmisión de la información entre los nodos para que el algoritmo propuesto funcione adecuadamente.



**Fig. 5.** Trama que trabajará en el prototipo de red 6LoWPAN

Esta trama es de gran utilidad al momento de monitorización en los logs enviados a la PC, para visualizar todos los procesos y finalmente la selección del nodo líder de respaldo. Y de esta manera asegurar que cada nodo termine de ejecutar un proceso correspondiente al algoritmo Omega para evitar conflictos al momento de ejecutar el algoritmo dentro del prototipo de red 6LoWPAN

La creación de la trama presentada en la Fig. 5, se ha implementado para la comunicación dentro de los nodos que van a participar en el algoritmo. A continuación, se detalla la explicación de cada campo de la trama.

1. Tipo de dato: Es un campo de 8 bits, que contendrá la información del tipo de dato que esté ejecutando cada nodo, esto se realizó en vista que cada nodo tiene asignado una ranura o slot de tiempo para transmitir sus datos, pero al

momento de tener varios nodos y varios procesos que conllevan al envío de transmisión de datos, como son como por ejemplo el caso del broadcast punto a punto desarrollado en la aplicación, la ejecución del algoritmo, el traspaso de mensajes; cada nodo que conforma el prototipo 6LowPAN debe ejecutar ordenadamente cada proceso (tipo de dato), para evitar conflictos y problemas en la ejecución del algoritmo. Por lo expuesto anteriormente, es necesario comprender que para poder pasar entre cada uno de los tipos de datos se debe previamente confirmar que todos los nodos inalámbricos hayan terminado de realizar un tipo de dato a la vez. Por esta razón se definió los siguientes tipos de datos en valor hexadecimal:

- ✓ 00: Este tipo de dato será de información y de visualización dentro del MoterRunner Shell, el cual permitirá al usuario ver la información pertinente a cada nodo inalámbrico.
- ✓ 01: Este tipo de dato solo será utilizado por el nodo líder seleccionado en un inicio dentro del prototipo de red de sensores inalámbricos 6LowPAN, éste a su vez va a realizar la petición del número máximo de nodos inalámbricos al nodo Gateway, esto como se explicó es uno de las adaptaciones, ya que el nodo que conoce acerca de toda la red 6LowPAN es el nodo Gateway. Para que así con esta información, cada uno de los nodos pueda ejecutar el algoritmo Omega, mismo que permitirá la selección del Nodo Líder de Respaldo.
- ✓ 02: Este tipo de dato será desarrollado en la aplicación para el proceso de Broadcast punto a punto, el mismo se encuentra optimizado, ya que al mismo tiempo de hacer un Broadcast para monitorear que cada nodo esté activo, también aprovecha este tipo mensaje para que cada nodo conozca su estado de batería y su identificador único que, ahora dentro del prototipo de red de sensores inalámbricos 6LowPAN, va a ser la dirección corta de cada nodo inalámbrico, que es una dirección de 8 bits.
- ✓ 03: Este tipo de dato permitirá el proceso de traspaso de mensajes entre todos los nodos inalámbricos del prototipo de red de sensores inalámbricos 6LowPAN. Con esto, cada nodo podrá seleccionar en base a su estado de batería y su dirección corta, el sucesor o nodo líder de respaldo que es el objetivo principal de este proyecto.
- ✓ 06: Este tipo de dato permite la confirmación del proceso correspondiente a cada tipo de dato y así afirmar que ha finalizado exitosamente en cada nodo inalámbrico, lo que equivale a que a cada nodo se le asignara un *Token* (turno), esto se realizó para que el paso de un tipo de dato a otro, con esto se pretende que todos los nodos que conforman la red 6LowPAN hayan terminado un proceso a la vez (Tipo de Dato).
- ✓ 07: Este tipo de dato permitirá el encendido del led Verde en el nodo inalámbrico seleccionado como Nodo Líder de respaldo, una vez que haya finalizado la ejecución del Algoritmo Omega. En el caso de los nodos que no hayan sido seleccionados como Nodo Líder de respaldo, el led indicador se apagará.

2. Número máximo de motas: Es un campo de 8 bits, que contendrá la información del número máximo de nodos inalámbricos que conforman el prototipo de red 6LowPAN, esto permitirá ejecutar el algoritmo en cada uno de los nodos inalámbricos.
3. Dirección corta: Es un campo de 8 bits, que contendrá la información de la dirección corta de cada nodo inalámbrico que conforman la red 6LowPAN, esto permitirá ejecutar el algoritmo en cada uno de los nodos y mediante la ejecución del algoritmo, se podrá determinar el nodo líder de respaldo en caso de existir dos o más nodos con el mismo estado de batería, en vista de que la aplicación en caso de encontrarse con esta situación, escogerá como nodo líder de respaldo a cuyo nodo contenga la dirección corta más alta.
4. Estado de la batería: Es un campo de 8 bits, que contendrá la información del estado o cantidad de batería de cada nodo inalámbrico que conforman la red 6LowPAN, esto permitirá ejecutar el algoritmo en cada uno de los nodos inalámbricos y además al final mediante la ejecución del algoritmo poder determinar el nodo líder de respaldo.
5. Dirección corta del sucesor: Es un campo de 8 bits, que contendrá la información del nodo seleccionado como nodo líder de respaldo de entre todos los nodos inalámbricos que conforman la red 6LowPAN, esto me permitirá corroborar el funcionamiento del algoritmo Omega en el prototipo de red 6LowPAN.

El paso de los mensajes con su debida trama estructurada, se envían utilizando el concepto de socket, cuya definición corresponde al conjunto formado por la dirección IP y un puerto lógico. Aquí ocurre una abstracción importante, ya que como se conoce, un puerto lógico normalmente pertenece a la capa Aplicación asociado a la dirección IP correspondiente. En este caso el protocolo MRV6 y la capa de adaptación 6LowPAN, creando el puerto a nivel de red y la dirección corta dentro de la red 6LowPAN, pero además el envío de los mensajes de confirmación se envían a una dirección de red IPv6, con esto se logra una total transparencia tanto en la red 6LowPAN con la red IPv6.

Dentro de la aplicación, se realiza básicamente la construcción de esta trama y para ser enviada, se utiliza el concepto de socket. Además, consta de la creación de dos socket el uno para el envío de información o de una manera más clara, es el que va a responder las peticiones realizadas por cualquier nodo que conforma el prototipo de red de sensores inalámbricos 6LowPAN; y, el segundo socket para poder recibir las peticiones hechas por los otros nodos inalámbricos de la red.

## 7 Análisis de Resultados

Para verificar que el algoritmo propuesto funciona correctamente, se hizo uso de los tres leds (rojo, verde) que disponen los nodos sensores, en especial del led verde, Al inicio todos los nodos tienen prendido el led verde, ya que inicialmente se consideran como potenciales Gateways de respaldo (nodos líderes) .

Al inicio todos los nodos ejecutan el algoritmo, cuando el nodo 4 finaliza la ejecución del algoritmo, envía una trama de finalización como se puede evidenciar en la Fig. 6,

este nodo envió su mensaje de encendido o apagado del led verde y a la vez de finalización. Con esto se consigue que los leds en los respectivos nodos que no se hayan seleccionado como líder se apaguen y únicamente el nodo que haya sido seleccionado como líder, encenderá el led verde.

De la misma manera cuando el último nodo inalámbrico del prototipo de red 6LowPAN haya terminado su proceso de encendido y apagado de leds, procederá a enviar una trama indicando que el proceso del algoritmo ha finalizado con éxito, como lo indica en la Fig. 5.

```

19:55:26.682'000 MRv6:INFO
Packet from note to note: 02-00-00-00-8E-07-28-1D -> 02-00-00-00-93-41-08-13
UDP: source:0|0|1|0:1028078E00000002 dest:0|0|1|0:130B419300000002 srcport:1023 dstport:1024 payload:0608000000
19:55:26.684'000 SONORAN:INFO
Media event not handled by any socket: category: note
evname: media
id: 27c
time: 19:55:26.682'000
dstport: 400
note: 02-00-00-00-8E-07-28-1D
srcport: 3ff
data: 00: 06 08 00 00 00
19:55:26.651'000 SONORAN:INFO
Media event not handled by any socket: category: note
evname: media
id: 280
time: 19:55:26.648'000
dstport: 5a
note: 02-00-00-00-93-41-08-13
srcport: 77
data: 00: 87
    
```

Fig. 6 Trama indicando la finalización del algoritmo Omega

Cuando dos nodos o más tengan el mismo estado de batería para ser elegido como nodo líder de respaldo, la selección se basará en el valor del identificador único más alto, que para el caso es la dirección corta de los nodos, esto es muy evidente ya que los nodos inalámbricos que conforman la red 6LowPAN se numeran desde el número 1 hasta el 4. Lo explicado anteriormente se lo puede apreciar en la Fig. 7.

```

19:44:53.271'000 MRv6:INFO
Packet from note for external address: 02-00-00-00-5D-F6-87-F7
UDP: source:0|0|1|0:F78F9500000002 dest:0|0|0|0:200000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004037300
19:44:59.561'000 MRv6:INFO
Packet from note for external address: 02-00-00-00-00-00-20
UDP: source:0|0|1|0:2000000000000002 dest:0|0|0|0:200000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004037300
19:44:59.566'000 MRv6:INFO
Packet from note for external address: 02-00-00-00-8E-07-28-1D
UDP: source:0|0|1|0:1028078E00000002 dest:0|0|0|0:200000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004037300
19:45:08.331'000 MRv6:INFO
Packet from note for external address: 02-00-00-00-5D-F6-87-F7
UDP: source:0|0|1|0:F78F9500000002 dest:0|0|0|0:200000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004037300
19:45:14.597'000 MRv6:INFO
Packet from note for external address: 02-00-00-00-8E-07-28-1D
UDP: source:0|0|1|0:1028078E00000002 dest:0|0|0|0:200000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004037300
19:45:23.378'000 MRv6:INFO
Packet from note for external address: 02-00-00-00-5D-F6-87-F7
UDP: source:0|0|1|0:F78F9500000002 dest:0|0|0|0:200000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004037300
19:45:29.651'000 MRv6:INFO
Packet from note for external address: 02-00-00-00-8E-07-28-1D
UDP: source:0|0|1|0:1028078E00000002 dest:0|0|0|0:200000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004037300
    
```

Fig. 7. Caso en el que dos nodos tengan el estado de batería más alto

Como se esperaba una vez ejecutado el algoritmo en cada nodo, es evidente que debía seleccionarse el nodo 4 como nodo líder de respaldo, en vista que el nodo 3 y el nodo 4 tenía el estado de batería más alto. Esto es lo que presenta la Fig. 8.



```

19:55:41.730'000 MRv6:INFO
Packet from mote for external address: 02-00-00-00-0E-D7-28-1D
UDP: source:0(0)1(0):F787F65000000002 dest:0(0)0(0):2000000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004047304

19:55:19.162'000 MRv6:INFO
Packet from mote for external address: 02-00-00-00-5D-F6-97-F7
UDP: source:0(0)1(0):F787F65000000002 dest:0(0)0(0):2000000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004037304

19:55:21.692'000 MRv6:INFO
Packet from mote for external address: 02-00-00-00-77-FF-34-3D
UDP: source:0(0)1(0):3034FF7700000002 dest:0(0)0(0):2000000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004013304

19:55:24.179'000 MRv6:INFO
Packet from mote for external address: 02-00-00-00-00-00-20
UDP: source:0(0)1(0):2000000000000002 dest:0(0)0(0):2000000000000000000000000000000011 srcport:1023 dstport:1024 payload:0004023004
  
```

Nodo 04  
seleccionado como  
nodo líder de  
respaldo

**Fig. 8** Selección el nodo líder con la dirección corta más alta

Un aspecto importante que hay que recalcar es que, la adaptación del algoritmo Omega dentro del prototipo de red 6LowPAN tiene un tiempo aproximado de doce minutos en terminar toda su ejecución, a pesar de que se trató de optimizar el código y reducir tiempos de procesamiento, se obtuvo este tiempo, ya que un limitante a ser considerado es el Hardware en sí, por la poca memoria disponible; además el uso limitado de programación y el tiempo en que demora cada nodo en procesar cada uno de los tipos de mensajes. Este tiempo se lo puede disminuir considerablemente si se considera a los nodos Gateways como nodos fijos y con alimentación de energía por la red eléctrica permitiendo la utilización de un hardware más potente.

## 8 Conclusiones y Futuros Trabajos

Al finalizar el proyecto descrito en este artículo, se ha implementado un prototipo de red de sensores inalámbricos 6LowPAN, en la cual se empleó dispositivos reales y se demostró la validez de la adaptación del algoritmo Omega en el prototipo de red 6LowPAN; teniendo como resultado el desarrollo de un algoritmo que mejora la confiabilidad de la red sensor inalámbrica, al seleccionar un Gateway de respaldo cuando el Gateway falla.

No hemos encontrado evidencia de trabajos similares que implementen el algoritmo Omega a nivel de red utilizando 6LowPAN utilizando el hardware de esta investigación, utilizando como métricas para la selección del Gateway de respaldo el nivel de Batería.

Como parte del prototipo, se ha implementado un conjunto de herramientas que ofrecen una solución completa tanto en software como en hardware, con las que fueron demostrados conceptos teóricos. En este contexto y considerando el uso de sensores inalámbricos 6LowPAN, una implementación de este tipo de redes, resulta ser una solución conveniente para ambientes IPv6 en diversas aplicaciones.

Previo a la implementación del prototipo, se estableció el escenario de pruebas para evaluar la adaptación del algoritmo Omega; los resultados de las pruebas permitieron establecer que la adaptación realizada fue compatible y se integró con el software realizado; sin embargo, presentó algunos inconvenientes, como son las limitantes en Hardware y Software las cuales deben ser sobrellevadas para el cumplimiento del objetivo de este proyecto.

Se concluye que el kit básico de desarrollo de redes WSN 6LowPAN Mote Runner de IBM y Libelium, es muy útil y adecuado para la comunidad de desarrolladores que recién están ingresando al mundo IPv6 y de redes de sensores inalámbricos. Además dicho kit provee módulos realmente útiles para el desarrollo de aplicaciones, así como ejemplos de los cuales se puede obtener gran aprendizaje.

Los resultados de ese algoritmo permitirán tener el nodo de respaldo de manera inmediata en la red 6LowPAN, reduciendo dramáticamente las pérdidas de información de los nodos, cuando el Gateway principal falla.

El proceso de la ejecución del algoritmo Omega utiliza funciones del protocolo IPv6, el mismo que fue adaptado a una red WSN 6LowPAN, considerando las limitantes tanto en hardware como en software.

Una vez finalizado este proyecto se corroboró el funcionamiento de la adaptación del algoritmo Omega en el prototipo de red de sensores inalámbricos 6LowPAN, además se observó que el tiempo que se tarda en ejecutar todo el algoritmo es alrededor de 12 minutos, si bien es un tiempo significativo, hay que tener en cuenta las limitantes del hardware, que como se conoce es un kit de desarrollo para iniciar en este tipo de redes de sensores inalámbricos, además todas las limitantes tanto de software como de hardware; sin embargo y a pesar de las restricciones impuestas, la aplicación ha sido desarrollada de la manera más óptima en términos de programación.

Se considera que los tiempos de ejecución del algoritmo pueden reducirse notablemente si en lugar de realizar los procesos a nivel de red, se los realiza con información del nivel de enlace, propuesta para futuras investigaciones.

Se recomienda que los desarrollos de nuevas aplicaciones incursionen y experimenten en otras temáticas en base a lo que ha sido desarrollado en este proyecto, que han sido mencionadas en este proyecto, ya que con la ayuda y el correcto manejo de redes de sensores inalámbricos pueden desarrollar aplicaciones de red que dan la capacidad de modificar y personalizar rápidamente el control de una WSN. Por lo tanto, se alienta a los desarrolladores a desarrollar aplicaciones que permitan que las redes de sensores inalámbricos sean tolerantes a fallas, es decir cuando exista alguna avería dentro de los nodos que conforman la red de sensores inalámbricos 6LowPAN, el impacto provocado no sea significativo. El presente proyecto deja un precedente para los desarrolladores que están experimentando tanto el mundo de IPv6 que va intrínsecamente enlazado con la evolución de redes de sensores inalámbricos.

La utilización de los algoritmos de enrutamiento IPV6 en la red 6LowPAN, considerando como métrica el nivel de batería, que permita identificar un Gateway óptimo de respaldo, es un trabajo a realizar.

## Agradecimientos

El presente artículo es uno de los resultados del proyecto semilla PIS 14-26 de la Escuela Politécnica Nacional

## Referencias

- [1] S. Chatterjea, P. Havinga, and S. Dulman, "Introduction to Wireless Sensor Networks," in *Embedded Systems Handbook*, 2005, pp. 10–31.
- [2] C. Ortega-Corral and L. E. P. Maestre, "6LowPAN, un paso inicial hacia el Internet de las 'cosas'," *Comput. Commun.*, p. 4, 2011.
- [3] A. C. Pinto, E. De la Hoz Franco, and D. C. Pinto, "Las redes de sensores inalámbricos y el Internet de las cosas," *INGE CUC*, vol. 8, no. 1, pp. 163–172, 2012.
- [4] G. J. Pottie, "Wireless sensor networks," *1998 Inf. Theory Work. (Cat. No.98EX131)*, no. 165, pp. 61–64, 1998.
- [5] M. Zennaro and I. Trieste-Italy, "Introduction to Wireless Sensor Networks," 2012.
- [6] C. Martín Hernández, "The omega failure in the crash-recovery model," Universidad del País Vasco, 2014.
- [7] S. Arévalo, E. Jiménez, M. Larrea, and L. Mengual, "Communication-efficient and crash-quiescent Omega with unknown membership," *Inf. Process. Lett.*, vol. 111, no. 4, pp. 194–199, 2011.
- [8] T. D. Chandra, V. Hadzilacos, and S. Toueg, "The weakest failure detector for solving consensus," *J. ACM*, vol. 43, no. 4, pp. 685–722, 1996.
- [9] E. Jiménez, S. Arévalo, and A. Fernández, "Implementing unreliable failure detectors with unknown membership," *Inf. Process. Lett.*, vol. 100, no. 2, pp. 60–63, 2006.
- [10] M. Larrea, C. Martín, and J. J. Astrain, "Fault-tolerant aggregator election and data aggregation in wireless sensor networks," *Int. J. Commun. Networks Distrib. Syst.*, vol. 3, no. 2, pp. 93–115, 2009.