

Uso de Redes Híbridas para la Mejora de la Conectividad en VANETs

Bernabé Dorronsoro, Patricia Ruiz, Grégoire Danoy, Pascal Bouvry y Lorenzo J. Tardón

Resumen—

Uno de los problemas más importantes existentes en las redes *ad hoc* es el particionamiento de la red, en el que se forman grupos aislados de manera que sólo dispositivos pertenecientes al mismo grupo pueden comunicarse. Ésta es una de las mayores limitaciones en la conectividad de una red. Normalmente todos los dispositivos que forman una red *ad hoc* incluyen interfaces de comunicación de corto alcance tipo WiFi o Bluetooth, además de otros que permitan la conexión con dispositivos remotos tales como GPRS/HSDPA. Esto nos permite la creación de redes híbridas al establecer enlaces entre dispositivos remotos (que llamaremos a partir de ahora *enlaces puente* o *bypass links* *BLs*) cuyo objetivo es mejorar la calidad de servicio de la red. Para colocar dichos *BLs* de forma óptima utilizaremos la teoría de mundo pequeño o *small world*, de forma que se maximice el coeficiente de agrupamiento en nuestra red y se minimice la longitud del camino característico. Además, estos enlaces puente (*BLs*) generalmente no son gratuitos, por lo que es necesario minimizar su uso. En este artículo se realiza un profundo estudio sobre el beneficio del uso de *BLs* para mejorar la conectividad de una red real optimizando tanto el número de *BLs* utilizado como su localización. Este estudio nos permitirá adquirir conocimientos para el diseño de mejores aplicaciones para el problema. Para validar la mejora que proporciona el uso de los *BLs*, se evalúa sobre la red resultante el comportamiento de dos protocolos: **BODYF**, un protocolo de difusión basado en una topología en árbol, y **Simple Flooding** un protocolo de inundación.

I. INTRODUCCIÓN

Últimamente, existe una tendencia en el campo de las redes *ad hoc* que se centra en el desarrollo de nuevas redes formadas por vehículos. En las redes *ad hoc* vehiculares, también conocidas como VANETs, los vehículos se pueden comunicar entre sí o con unidades de carretera que permiten el acceso a un sistema final capaz de proporcionar alertas, información del tráfico, etc. La comunicación entre vehículos es un tema de investigación que está tomando mucho interés y existen numerosos proyectos tanto nacionales como internacionales [4], [5], [12] trabajando en ello. Las aplicaciones harán que la conducción sea más segura y más eficiente. Algunos ejemplos son las aplicaciones de alertas, como por ejemplo, permitir que los vehículos envíen mensajes avisando de un peligro, del estado del tráfico o de las carreteras, etc. En VANETs, debido a la alta velocidad de los dispositivos, la topología de la red es incluso más dinámica que en las MANETs, lo cual dificulta la comunicación.

Univ. de Luxemburgo. E-mail: bernabe.dorronsoro@uni.lu.
Univ. de Luxemburgo. E-mail: patricia.ruiz@uni.lu.
Univ. de Luxemburgo. E-mail: gregoire.danoy@uni.lu.
Univ. de Luxemburgo. E-mail: pascal.bouvry@uni.lu.
Univ. de Málaga. E-mail: lorenzo@ic.uma.es.

Las VANETs, y normalmente todas las redes *ad hoc* multi-salto, se enfrentan al problema del particionamiento de la red. En este trabajo, se considera el problema de optimizar la conectividad de estas redes añadiendo enlaces de largo alcance (utilizando por ejemplo tecnologías como GPRS/HSDPA). Estos enlaces de largo alcance son conocidos como enlaces puente o *bypass links* (*BLs*) y pueden utilizarse para conectar distintas particiones de una red. Además, estos *BLs* no son gratuitos, por lo que hay que optimizar el uso que se hace de ellos. En este trabajo planteamos un profundo estudio del problema que nos permitirá adquirir conocimientos para el desarrollo de futuras soluciones realistas. Para ello, utilizaremos algoritmos genéticos celulares (*cGAs*) [1] con el fin de encontrar soluciones cercanas a las óptimas. El objetivo es minimizar el número de *BLs* a utilizar, optimizando su localización y mejorando, por tanto, la conectividad de la red.

El escenario en el que se va a trabajar consiste en el centro de la ciudad de Luxemburgo. El motivo para escoger este entorno es que el centro está cubierto por puntos de acceso repartidos por toda la ciudad, lo que nos permitirá tener conocimiento global de nuestro sistema. De esta forma, podemos contar con servidores que tienen conocimiento global de la red, que permitirán calcular la localización óptima de estos *BLs*. En este trabajo se consideran las propiedades de mundo pequeño para evaluar la calidad de nuestra solución [16].

La contribución de este artículo es triple. Por un lado, abordamos el problema de la conectividad en redes móviles *ad hoc* en las que los dispositivos se mueven a gran velocidad (VANETs) y se analizan los resultados obtenidos. Por otro lado, se realiza un profundo estudio en la influencia que tiene el número de *BLs* utilizado en el comportamiento de la red sobre tres densidades de red distintas. Además, para cada densidad de red se utilizan 5 topologías diferentes. Como tercera contribución, además de evaluar numéricamente la conectividad de la red mediante nuestra función de fitness, el comportamiento de la red se analiza experimentalmente en función del número de dispositivos que son capaces de recibir un mensaje de difusión o *broadcast* (dos protocolos de difusión se han utilizado para ello).

El resto del artículo se organiza como sigue. En la siguiente sección se introduce el problema de la optimización de la conectividad de la red utilizando *BLs*. La Sección III describe los protocolos de difusión utilizados, y en la Sección IV se presenta la

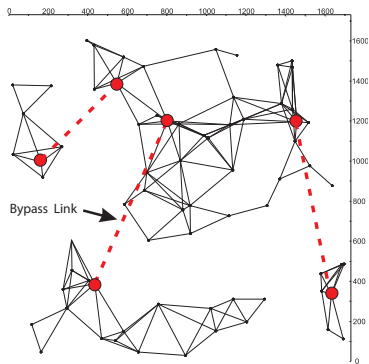


Fig. 1. Uso de los enlaces puente.

estrategia propuesta para afrontar el problema. Los experimentos realizados y los resultados obtenidos se muestran en la Section V y, finalmente, presentamos nuestras conclusiones en la última sección.

II. ESTUDIO GENERAL DEL PROBLEMA

Básicamente, el problema que se estudia en este artículo es el de solventar el particionamiento de la red en VANETs mediante la localización de BLs. Además, estos BLs se colocarán de forma que se optimice la conectividad de la red y se minimice a la vez el número de BLs utilizados.

La motivación inicial para adentrarnos en esta línea de investigación fue la suposición de que tecnologías como Bluetooth y WiFi se pueden utilizar para crear enlaces de comunicación en modo *ad hoc* gratuitos, pero además se pueden emplear enlaces de redes celulares tales como GPRS/HSDPA [9], [10], [14] en los dispositivos que los soporten, de manera que se establezcan enlaces complementarios llamados enlaces puente o *bypass links* (BLs). Estos enlaces tendrán un coste adicional y se utilizan para comunicar dispositivos distantes (que no están en rango de cobertura), bien porque requieren muchos saltos de comunicación entre ellos o porque pertenecen a particiones distintas de la red. En la práctica, un enlace puente se puede crear utilizando una conexión de la red celular o utilizando los puntos de acceso. De cualquier manera, en este modelo se considera un enlace puente como un sólo salto de comunicación para simplificar la topología real que hay tras el *bypass link* (véase el ejemplo de la Figura II).

En este entorno, como ya se mencionó, existen puntos de acceso distribuidos por todo el escenario, capaces de tener información global de todo el sistema. Será en estos servidores en los que se calculará y optimizará tanto el número de BLs a utilizar como los dispositivos que se enlazarán. Para comprobar el rendimiento de las nuevas redes con los BLs, se va a estudiar la cobertura que alcanzan dos protocolos de difusión antes y después de utilizar estos enlaces puente. Estos dos protocolos son BODYF y SF. Por un lado SF [11], [17] es una técnica de inundación que no utiliza ningún tipo de conocimiento de la red, y que tampoco intenta reducir el número de reemisiones o de colisiones durante el proceso de difusión. SF es instantáneo y llega a todos los dispositivos dentro

de una misma partición. Por otro lado, BODYF [13] es un protocolo de difusión que utiliza topología en árbol y que hace uso de la información del vecindario para minimizar el número de reenvíos necesarios, a la vez que trata de maximizar el número de dispositivos que recibirán el mensaje. Aunque BODYF fue diseñado para hacer frente al particionamiento de la red y diseminar el mensaje más allá de la partición del nodo emisor, es un reto muy complicado conseguir una cobertura elevada dada la gran velocidad y movilidad de los dispositivos. Por tanto, en este trabajo se utilizará un cGA para encontrar el número correcto de BLs y sus posiciones de forma que se maximice el número de dispositivos que finalmente recibirán el mensaje. El criterio para determinar los dispositivos que van a formar los BLs puede influir altamente en el comportamiento de la red.

Para abordar este problema, se utilizan las propiedades de mundo pequeño o *small world* [16] como indicadores de la calidad de nuestra solución. Las redes *small world* se caracterizan por tener un alto coeficiente de agrupamiento (γ) y a la vez mantienen una longitud de camino característico baja (L). Por un lado, tener una longitud de camino característico baja es importante para tener mecanismos de enrutamiento efectivos y también para el rendimiento de la comunicación global de la red. Minimizar la longitud del camino característico significa minimizar el número de saltos necesarios para comunicar dos dispositivos cualesquiera. Por otro lado, tener un coeficiente de agrupamiento alto implica una gran conectividad en el vecindario de cada nodo y, por tanto, cada nodo puede alcanzar un alto grado de difusión de información. La principal motivación para utilizar las propiedades del mundo pequeño, es que este tipo de redes combinan las ventajas de las redes regulares (alto coeficiente de agrupamiento) con las ventajas de las redes aleatorias (longitud de camino característico baja). Para optimizar estos parámetros (maximizar γ , minimizar L) y minimizar el número de BLs en la red, utilizamos algoritmos evolutivos (EAs) [3], más específicamente algoritmos genéticos celulares (cGAs) [1].

III. ALGORITMOS DE DIFUSIÓN CONSIDERADOS

Nos centramos en los protocolos de difusión porque el *broadcasting* es una de las operaciones fundamentales a bajo nivel en la que se basan muchas aplicaciones de más alto nivel, como el enrutamiento en redes móviles ad hoc. En MANETs, el hecho de que los dispositivos sean móviles y que tengan un radio de cobertura limitado, provoca que muchos dispositivos en un determinado momento no estén al alcance de otros y que la topología sea altamente fluctuante. Por esta razón, muchos investigadores se están centrando en la optimización del comportamiento de estos algoritmos, maximizando el número de nodos alcanzados y minimizando tanto el tiempo necesario como la carga de la red [1], [2].

Con el fin de hacer autocontenido este artículo, presentamos a continuación una breve descripción de los dos protocolos de difusión con los que hemos medido la calidad de la solución propuesta: BODYF y simple flooding (SF). SF ha sido elegido porque es un protocolo que produce una difusión instantánea del mensaje por los dispositivos de la partición, por lo que puede utilizarse como referencia para evaluar la calidad de las redes encontradas por el cGA. Por otro lado, BODYF es un protocolo mucho más avanzado y apropiado para redes reales que trata de minimizar el uso de la red, y que se aprovecha de la movilidad de los dispositivos para diseminar el mensaje a otras particiones distintas de aquella en la que se comenzó el *broadcast*.

A. El Protocolo de Difusión BODYF

BODYF es un protocolo de difusión (o *broadcasting*) específico para redes de comunicación dinámicas basadas en topologías de árboles de expansión. Este tipo de topologías es usualmente necesario para las capas superiores, tal vez para hacer enrutamiento o cualquier otra aplicación [20]. Por tanto, BODYF trata de alcanzar la máxima cobertura en el menor tiempo posible a un mínimo coste, utilizando la información de red que nos proporciona la topología en árbol. Esta topología de árbol se puede crear y mantener en la red de forma descentralizada utilizando, por ejemplo, el modelo DAGRS [18], [19]. Este modelo se basa en cuatro reglas muy simples con las que se toman decisiones en función del estado de los dispositivos. Para conseguir una topología en árbol, es decir, un grafo conexo y sin ciclos, existe un token circulando por él, de manera que sólo aquel dispositivo que tenga el token podrá tomar decisiones.

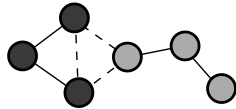


Fig. 2. Tipos de vecinos en una red con topología en árbol.

Asumiendo que tenemos el árbol (o bosque debido a las particiones) establecido en la red, podemos diferenciar entre dos tipos de vecinos: (1) vecinos lógicos, que son aquellos que pertenecen al mismo árbol, y (2) vecinos potenciales, que son aquellos que están en rango, pero que bien no pertenecen al mismo árbol o bien pertenecen al mismo árbol pero no están conectados para evitar que se formen ciclos (ver Figura 2). En esta imagen, los dispositivos del mismo color pertenecen al mismo árbol. Los enlaces entre los dispositivos se representan mediante una línea continua si pertenecen al mismo árbol (vecinos lógicos) y discontinua cuando son vecinos potenciales.

El pseudocódigo de BODYF se presenta en el Algoritmo 1. En él se explica el comportamiento del protocolo. Para BODYF, todos los mensajes tienen un identificador único, por lo tanto, un dispositivo

Algoritmo 1 Pseudocódigo de BODYF.

Dato: m : el mensaje entrante.

Dato: d : el nodo que recibe el mensaje m .

Dato: s : el nodo que envió m .

```

1: if  $m$  se recibe por primera vez then
2:   if  $s$  y  $d$  pertenecen al mismo árbol then
3:      $d \rightarrow$  reenvía  $m$ ;
4:   else
5:     espera hasta recibir el token  $\rightarrow$   $d$  tiene el token;
6:   if  $d$  recibió  $m$  por su mismo árbol then
7:      $d \rightarrow$  descarta  $m$ ;
8:   else
9:      $d \rightarrow$  reenvía  $m$ ;
10:  end if
11: end if
12: else
13: descarta  $m$ 
14: end if

```

es capaz de detectar si el mismo mensaje le llega más de una vez. Cuando esto sucede, el mensaje se descarta directamente y no se hace ningún tipo de procesamiento. Si un dispositivo recibe por primera vez un mensaje, existen dos posibilidades: (1) el mensaje procede de un vecino lógico, o (2) procede de uno potencial. En (1), el dispositivo lo reenvía inmediatamente, en otro caso, el dispositivo espera hasta recibir el token (línea 5). Una vez que el nodo recibe el token, éste reenviará el mensaje si y sólo si durante el tiempo en el que estuvo esperando el token no recibió el mismo mensaje procedente de un vecino que pertenece a su mismo árbol (líneas 6-10), si ya fue recibido lo descarta.

B. El Protocolo de Inundación Simple Flooding, SF

El protocolo de inundación, más conocido como *Simple Flooding (SF)*, hace uso de la idea más intuitiva para diseminar un mensaje: cada nodo reenvía el mensaje una sola vez. SF no intenta disminuir el número de reemisiones ni de colisiones, y además no requiere conocimiento de ningún tipo sobre la red o el vecindario. Esta técnica no es nada recomendable cuando el número de dispositivos en la red es muy alto. En redes muy densas provocaría mucho tráfico en la red, e incluso podría llegar a congestionarla. Este problema se conoce como el problema de la tormenta en difusión (*broadcast storm problem*) [11].

IV. NUESTRA PROPUESTA

Como se ha comentado anteriormente, para resolver el problema propuesto se utiliza un algoritmo de población estructurada llamado algoritmo genético celular (cGA).

El uso de poblaciones descentralizadas en GAs normalmente lleva a algoritmos más eficientes (encuentran la solución con menor esfuerzo), más efectivos (encuentran soluciones más frecuentemente) y más precisos (la calidad de las soluciones es mejor), que GAs con poblaciones centralizadas [1].

En esta sección se presenta una breve descripción de los cGAs (Section IV-A), la representación del problema y también la función de evaluación que se utiliza para resolver el problema (Sección IV-B).

A. Algoritmos Genéticos Celulares

Los algoritmos genéticos celulares (cGAs) [1] son un tipo de algoritmo genético (GA) con población estructurada en la que los individuos se reparten en una red toroidal bidimensional, y sólo pueden relacionarse con sus vecinos. Por ejemplo, en la Figura 3 se muestra la disposición de individuos en la población de un cGA, el vecindario del individuo central (sombreado), y de otro vecino alejado del centro en la parte superior izquierda (línea discontinua).

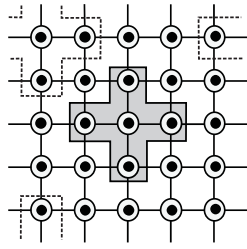


Fig. 3. Población de 5×5 de un cGA con vecindario L5.

En el Algoritmo 2 se incluye el pseudocódigo de un cGA canónico. Normalmente, en este cGA básico, la población se estructura en una rejilla de dimensión d ($d = 1, 2, 3$), y sobre ésta se define el vecindario. El algoritmo considera, de manera iterativa, como individuo actual a todos los individuos en la rejilla (línea 3), y éstos sólo pueden relacionarse con individuos que pertenezcan a su vecindario (línea 4), por lo que los padres se escogen entre los vecinos (línea 5) según un criterio determinado. Los operadores de cruce y mutación se aplican a los individuos en las líneas 6 y 7, con probabilidades P_c y P_m , respectivamente. Después, el algoritmo calcula el valor de adecuación o valor de *fitness* del nuevo individuo generado (línea 8), y lo inserta en lugar del individuo actual de la población (línea 9) siguiendo una determinada política de reemplazo. Este bucle se repite hasta que se alcanza la condición de terminación (línea 2).

Algoritmo 2 Pseudocódigo de un cGA canónico.

```

1: proc Evoluciona(cga) //Parám. del algoritmo en 'cga'
2: while ! CondiciónDeParada() do
3: for individuo  $\leftarrow$  1 to cga.TamañoPoblación do
4: n_lista  $\leftarrow$  Obtener_Vecindario(cga, posición(individuo));
5: padres  $\leftarrow$  Selección(n_lista);
6: descendiente  $\leftarrow$  Recombinación(cga.ProbRecomb, padres);
7: descendiente  $\leftarrow$  Mutación(cga.ProbMut, descendiente);
8: Evaluación(descendiente);
9: Añadir(posición(individuo), descendiente, cga);
10: end for
11: end while
12: end proc

```

B. Representación del Problema y Función de Evaluación

La representación de la solución es un aspecto fundamental en este tipo de algoritmos ya que determinará el tamaño del espacio de búsqueda y la elección de los operadores genéticos que se aplicarán para explorarlo. Aquí se ha utilizado una codificación entera de la solución en la que cada gen se corresponde con un posible enlace puente de todos los que se podrían formar. Por ejemplo, si se ha fijado previamente el número máximo de BLs de la red a 10, entonces un cromosoma tendrá 10 genes. La Figura 4 muestra el ejemplo de un cromosoma compuesto por 2 genes (número máximo de BLs permitidos es 2) en una red que tiene 5 nodos. La mitad superior derecha de la matriz de 5×5 representa todos los BLs posibles de la red, incluyendo los posibles enlaces locales que ya existen (debido a conexiones WiFi existentes en modo *ad hoc*). En el ejemplo mostrado en la Figura 4, el primer gen marcado con un círculo, con valor 2, representa una conexión entre los dispositivos 1 y 3 en la correspondiente matriz superior derecha, también marcada.

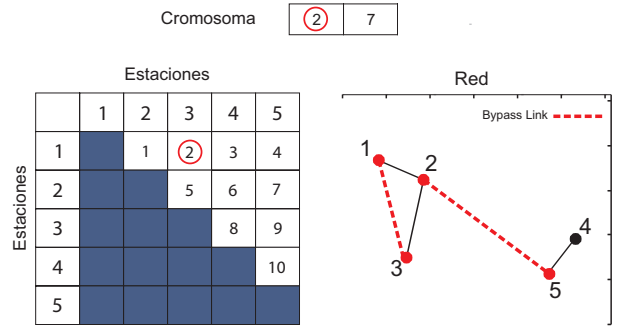


Fig. 4. Ejemplo de la representación de la solución.

Para asignar un valor de adecuación a las soluciones candidatas, es decir conjunto de posibles BLs, de nuestro algoritmo, utilizamos una única función de coste F , que combina las dos medidas que teníamos de mundo pequeño (L y γ) con el número de *bypass links*. El cálculo de la longitud del camino característico L impone que exista un camino entre dos nodos cualesquiera a y b . Consecuentemente, el cálculo de la función de evaluación requiere que primero se evalúe si la red está particionada. Si la red optimizada sigue teniendo particiones (los BLs definidos no consiguen conectar todas las particiones), se considera que el valor de *fitness* es un valor ponderado del número de particiones que todavía tiene la red. De esta forma, el algoritmo tratará de minimizar el número de particiones en caso de que no se encuentre una red totalmente conexa. Por otro lado, si la red optimizada no tiene particiones, el valor de adecuación es una combinación lineal del coeficiente de agrupamiento, de la longitud característica del camino y de la diferencia entre el número de BLs definidos y el número máximo de BLs permitidos.

El objetivo del proceso de optimización es maximizar el coeficiente de agrupamiento y minimizar tanto la longitud característica del camino como el número de BLs. Por tanto, estamos ante un problema multi-objetivo [6]. De cualquier forma, en este trabajo hemos utilizado una función de evaluación, definida en [21] y descrita en el Algoritmo 3, mediante la que se modela el problema como una única función a maximizar. La razón es que nuestro sistema necesita una única solución para aplicar a la red. Se podría también considerar la resolución del problema multi-objetivo, pero en este caso tendríamos que diseñar un segundo algoritmo que eligiera de alguna forma la más apropiada de las soluciones encontradas.

Algoritmo 3 Función de evaluación.

```

1: if Grafo conexo then
2:    $F = \alpha * \gamma - \beta * (L - 1) - \delta * (bl - bl_{max})$ 
3: else
4:    $F = \xi * P$ 
5: end if

```

En el Algoritmo 3, bl es el número BLs creados por una solución en la red simulada, bl_{max} (definido previamente) es el número máximo de BLs que se pueden crear en la red, P es el número de particiones restantes después de añadir los BLs y N es el número de estaciones en el sistema global. Finalmente, hemos tomado los pesos utilizados en [7] para las siguientes variables (ConexionesWiFi es el número de conexiones WiFi existentes en la red):

$$\alpha = 1$$

$$\beta = 1 / (N - 2)$$

$$\delta = 2 / (N * (N - 1)) - \text{ConexionesWiFi}$$

$$\xi = -1,0$$

V. EXPERIMENTOS

En esta sección se describe el proceso seguido para realizar los experimentos y se presentan y analizan los resultados obtenidos. Tras localizar los BLs que hacen máxima la conectividad de la red evaluamos su comportamiento mediante la difusión de un mensaje, utilizando tanto BODYF como SF. En la Sección V-A se describe brevemente el simulador y el modelo de movilidad utilizado. Los parámetros utilizados para el cGA en los experimentos realizados se presentan en la Sección V-B. Finalmente, los resultados se muestran en la Sección V-C.

A. Descripción del Modelo de Movilidad

En este trabajo, hemos evaluado nuestras soluciones en el entorno de las redes *ad hoc* vehiculares (VANETs). Por lo tanto, se considera que cada dispositivo es un automóvil y que es capaz de comunicarse con automóviles vecinos. El escenario utilizado para nuestras simulaciones es el centro de la ciudad de Luxemburgo. La Figura 5 muestra un ejemplo de una VANET simulada en nuestro escenario, que además hace uso de varios enlaces puente para mejorar la conectividad de la red.

Las redes estudiadas son muy similares a la presentada en la Figura 5. Para nuestras pruebas generamos tres tipos de densidades de red distintas y 5 topologías en cada una de estas densidades, de manera que los resultados obtenidos sean fiables. Para intentar que la simulación sea lo más realista posible, utilizamos un modelo de movilidad creado específicamente para este escenario [13], en el que los dispositivos tienen los movimientos restringidos (sólo se mueven a lo largo de las calles), se paran en los cruces, pueden adelantarse unos a otros, etc. El área de simulación es de $1323\text{m} \times 1863\text{m}$, lo que equivale a unos 2.465 Km^2 .

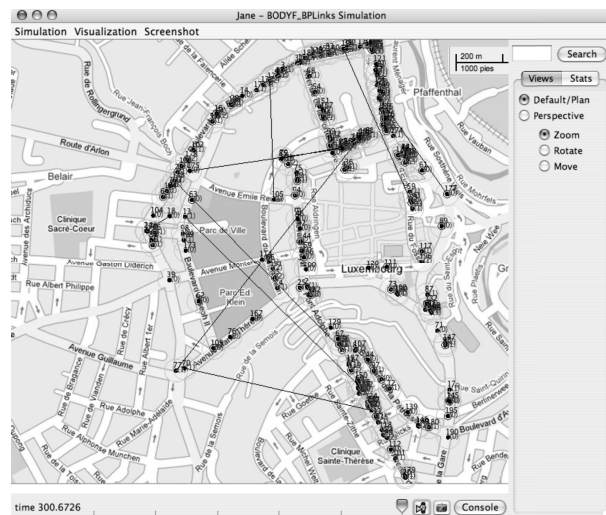


Fig. 5. Modelo de movilidad en el centro de Luxemburgo.

Los parámetros utilizados en este escenario son los mismos que los utilizados en [13]. La velocidad de los dispositivos varía entre 8,5 y 21 m/s. La cobertura de los mismos es de 100 metros. Los dispositivos se mueven en línea recta de un cruce a otro a una velocidad constante. De esta forma conseguimos un modelo muy realista, ya que incluso las áreas más congestionadas de Luxemburgo en cuanto al tráfico se reflejan en este modelo. En esta situación la dificultad para diseminar un mensaje es mayor que si usamos *random waypoint*, ya que no es nada sencillo llegar a algunas zonas al tener los movimientos más restringidos. Pero sin lugar a dudas, la simulación es mucho más realista.

Para simular estos escenarios hacemos uso de JANE [8], [15]. La principal característica de JANE es su desarrollo en tres etapas; facilita la evaluación y minimiza el esfuerzo necesario para desarrollar software para MANETs, de manera que el código evaluado en el entorno simulado, pueda ser ejecutado directamente en escenarios reales sin necesidad de modificarlo.

Tras introducir brevemente tanto el simulador como el entorno de simulación que hemos diseñado para este trabajo, pasamos a explicar el proceso que hemos seguido en nuestros experimentos. En primer lugar, dejamos que el simulador evolucione la red en

el escenario durante 300 segundos con el fin de estar seguros de que los dispositivos están distribuidos de forma realista por el escenario. Este paso es necesario porque inicialmente JANE coloca a los dispositivos en los distintos cruces del escenario de forma aleatoria, por lo que se encuentran formando clusters. Tras los 300 segundos de simulación, almacenamos la topología de la red en una matriz, que será utilizada como dato de entrada por nuestro algoritmo de optimización para buscar el conjunto de BLs que optimice la conectividad de la red. Tras esto, añadimos a la red las nuevas conexiones representadas por los BLs encontrados por el cGA y diseminamos el mensaje de *broadcast*. Como mecanismos de difusión del mensaje, se utilizan los protocolos BODYF y SF como ya se mencionó con anterioridad. El motivo por el que se escoge BODYF es porque mejora la cobertura conseguida por otros protocolos en este tipo de redes [13]. SF se escoge como referencia, ya que se sabe que alcanza todos los dispositivos en una partición, pero no es capaz de diseminar el mensaje más allá de ésta. Por lo tanto, comparar los resultados obtenidos con SF utilizando y sin utilizar BLs, puede dar una cierta medida de la calidad de la conectividad en la red resultante tras añadir los BLs. Con el fin de que las comparaciones sean justas, el nodo desde el que comienza la diseminación del mensaje es el mismo para los dos protocolos y la misma topología, densidad, y número máximo de BLs permitidos.

B. Configuración

En la Tabla I se muestran los parámetros de nuestro cGA. Se ha utilizado una única población de 100 individuos, y en la que los individuos se sitúan en una red cuadrada de 10×10 . El vecindario utilizado es L5 (también llamado NEWS o Linear5), de forma que se limitan las relaciones de los individuos a sus vecinos directos del norte, sur, este y oeste. Uno de los padres es el propio individuo, mientras que el otro se selecciona de entre el vecindario utilizando torneo binario (BT). Esto significa que se seleccionan dos individuos aleatoriamente, y el mejor (el que tenga mejor valor de adecuación) será seleccionado. Los dos padres se recombinan utilizando el operador de cruce de dos puntos (DPX), aplicado con una probabilidad $p_c = 0,8$. DPX consiste en la elección aleatoria de dos puntos de corte en los cromosomas de los padres y se escoge la mayor parte del cromosoma del mejor padre y la parte más pequeña del peor padre, construyendo así el nuevo descendiente, véase Figura 6.

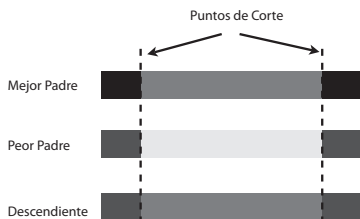


Fig. 6. Operador de cruce de dos puntos.

TABLA I
PARÁMETROS UTILIZADOS PARA EL ESTUDIO DEL cGA.

Tamaño de la población	10×10
Vecindario	L5
Selección	individuo actual + BT
Operador de cruce	DPX, $p_c = 0,8$
Operador de mutación	un gen, $p_m = 1/\text{longitud_crom.}$
Reemplazo	reemplaza si no es peor
Condición de parada	50000 evaluaciones de la función

El operador de mutación utilizado es un operador de mutación de un gen, que consiste en cambiar el valor de un gen por otro valor escogido de manera aleatoria, y que se aplicará a cada gen con una probabilidad de $p_m = 1/\text{longitud_cromosoma}$. Finalmente, el algoritmo parará cuando alcance 50000 evaluaciones de la función de evaluación.

Con el objetivo de obtener valores fiables, se han estudiado tres tipos de densidades y 5 topologías distintas para cada densidad de red. Los diferentes escenarios estudiados son: (1) una red de 500 dispositivos, (2) otra un poco más densa de 700 y finalmente, (3) una tercera con 1000 nodos.

C. Resultados

En esta sección se comparan los resultados obtenidos utilizando 0, 10, 20, 30 y 50 BLs, en las redes de 500 y 700 dispositivos, con 5 topologías para cada densidad. Por tanto, se consideran 50 redes distintas, lo que implican 1100 experimentos, ya que cada una de estas redes se resuelve 20 veces por el cGA. Esto es para cada uno de los protocolos, BODYF y SF. El cGA se ejecuta 20 veces para cada topología de red dada su naturaleza estocástica, ya que distintas ejecuciones darán lugar probablemente a distintos resultados. Aquí se considera la mejor de las 20 soluciones encontradas para cada protocolo. Finalmente, se extendió este trabajo realizando otras pruebas con redes de 1000 dispositivos utilizando en este caso 4 topologías distintas con 0, 10 y 20 BLs. En este último caso en lugar de 20 ejecuciones sólo se hacen dos dado que el cGA emplea mucho tiempo en resolver el problema (una media de 87,47 horas para el caso de los 20 BLs). No se consideran más de 20 BLs dado que con este número ya se encuentran redes completamente conexas.

La calidad de las redes resultantes se evalúa en términos de la cobertura alcanzada por cada uno de los protocolos, comparándola con la cobertura que tiene sin utilizar BLs. Las Tablas II y III muestran la media y la desviación estándar del número de dispositivos alcanzados por ambos protocolos, utilizando un máximo de 0, 10, 20, 30 y 50 BLs para las redes de 500 y 700 dispositivos.

El primer experimento, con densidad de 500 dispositivos, se considera que es una red muy dispersa, por lo que el número de particiones es muy elevado y por lo tanto es muy difícil diseminar el mensaje incluso utilizando BLs. Como se puede observar en la

TABLA II
RESULTADOS MEDIOS OBTENIDOS CON SF.

	sin BLs	No. BL	con BLs
500 disp.	84,20 \pm 66,42	10	168,00 \pm 98,15
		20	262,60 \pm 96,45
		30	282,80 \pm 136,64
		50	424,80 \pm 31,75
700 disp.	179,20 \pm 139,77	10	337,60 \pm 120,76
		20	330,40 \pm 176,43
		30	521,00 \pm 44,64
		50	641,60 \pm 27,61

TABLA III
RESULTADOS MEDIOS OBTENIDOS CON BODYF.

	sin BLs	No BL	con BLs
500 disp.	128,40 \pm 114,16	10	243,60 \pm 145,31
		20	318,20 \pm 161,98
		30	434,00 \pm 41,40
		50	464,80 \pm 4,55
700 disp.	405,60 \pm 201,45	10	589,00 \pm 38,24
		20	620,00 \pm 24,42
		30	648,80 \pm 47,46
		50	664,40 \pm 16,88

Tabla II, el número de dispositivos alcanzados con SF aumenta conforme se le añaden BLs. Especialmente, se debe prestar atención al incremento en el número de dispositivos cuando se pasa de no utilizar enlaces puente a usar 10 BLs, ya que casi se duplica la cobertura conseguida.

En la Tabla III se observa en BODYF el mismo efecto que en SF: se consigue llegar casi al doble de dispositivos utilizando tan sólo 10 BLs. Este dato es muy interesante ya que en una red de 500 dispositivos el hecho de utilizar 10 BLs significa que sólo el 4 % utilizarían enlaces puente, si la red es de 700 o 1000 dispositivos serían un porcentaje inferior. Al igual que en SF, como era de esperar, el número de dispositivos alcanzados aumenta al incrementar el número de BLs. Pero se puede observar como el efecto de pasar de 30 a 50 BLs no supone una gran mejora de la cobertura.

En el caso de 50 BLs, en la red de 500 dispositivos, el cGA es capaz de encontrar una red totalmente conexas utilizando 40, 43 y 44 BLs, pero siempre en la misma topología. Observamos que ninguno de los dos protocolos es capaz de llegar a la totalidad de dispositivos. BODYF llega a 466 y SF a 421. En realidad este número se acerca mucho al valor medio que se obtiene de todas las topologías. Esto significa que incluso cuando la solución que encuentra el cGA no es perfecta (la red no es totalmente conexas), el número de dispositivos alcanzados en ambos protocolos es bastante cercano al óptimo. El hecho de no llegar a todos los dispositivos puede ser debido a la alta velocidad con la que se mueven los dispositivos (hasta 21 m/s), lo cual provoca que la topología de la red cambie rápidamente, haciendo que aparezcan nuevas particiones en la red en muy poco tiempo.

Si prestamos atención ahora a la red de 700 dispositivos, se observan resultados similares al caso de 500 dispositivos. El uso de 10 BLs provoca duplicar la cobertura en SF, y aumentar el uso de BLs implica, igualmente, un aumento del número de dispositivos alcanzados.

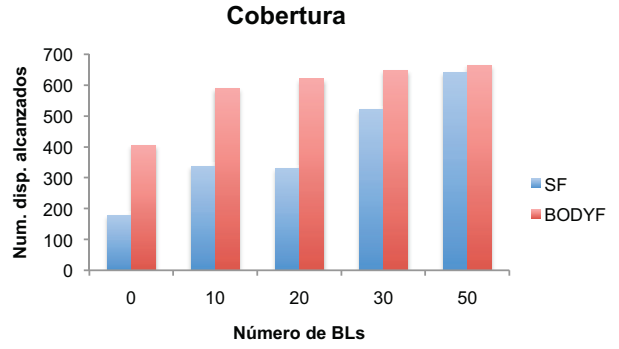


Fig. 7. Cobertura de BODYF y SF en una red de 700 disp.

positivos alcanzados. En el caso de BODYF, se sigue aumentando la cobertura al incrementar el número de BLs. Sin embargo, en esta red más densa la diferencia entre utilizar 10 BLs o 50 BLs no es tan grande (el 10 % del número total de dispositivos). Se puede observar cómo al aumentar la densidad de la red la diferencia en los resultados con los diferentes BLs se reduce en el caso de BODYF. De hecho, este efecto se podría explicar con el buen comportamiento de BODYF al diseminar mensajes en una red móvil *ad hoc* particionada, ya que el número de dispositivos alcanzados sin utilizar BLs ya es elevado (405.60 dispositivos en media frente a los 179.20 en el caso de SF).

En la Figura 7 se muestra la evolución de la cobertura de ambos protocolos en una red con 700 dispositivos. Para BODYF vemos un gran salto en la cobertura entre las redes de 0 y 10 BLs, pero sin embargo, vemos que utilizar más de 10 BLs no reporta grandes beneficios. Esto pone de manifiesto la eficiencia de BODYF para superar las particiones de la red, teniendo en cuenta que con un número menor de BLs, el porcentaje de la cobertura obtenida ya es muy alto.

Tras el análisis de estos resultados, se decidió estudiar también el caso de redes con 1000 dispositivos, y comparar los resultados sin BLs, con 10 y con 20 BLs. No se considera un número mayor de BLs, porque el cGA ya encuentra red conexas para todas las topologías utilizando sólo 20 BLs. Los resultados se presentan en la Tabla IV, y confirman la tendencia observada con las otras densidades de red. Por un lado, la cobertura crece al incrementar el número de BLs en SF. Por el otro, la diferencia entre usar 10 o 20 BLs se reduce considerablemente al usar BODYF. Es importante mencionar en este punto que la cobertura conseguida con BODYF utilizando 10 y 20 BLs es realmente alta: 90.65 % y 97.68 % de los dispositivos que forman la red, respectivamente.

En la Figura 8 se muestra la evolución de la cobertura para los distintos tipos de redes conforme aumentamos el número de BLs. Se puede observar la tendencia comentada anteriormente para BODYF al aumentar el número de BLs. A partir de un número determinado de BLs, obtenemos que incrementar el

TABLA IV
RESULTADOS MEDIOS OBTENIDOS CON BODYF Y SF PARA
UNA RED DE ALTA DENSIDAD (1000 DISP).

	sin BLs	No. BL	con BLs
SF	672,0 \pm 102,1	10	793,8 \pm 101,7
		20	892,8 \pm 45,0
BODYF	452,5 \pm 312,0	10	906,5 \pm 44,3
		20	976,8 \pm 11,5

número de enlaces puente utilizados no reporta importantes variaciones en la cobertura del protocolo. Cuanto más densa es la red, menos BLs son necesarios para llegar a esa zona de la gráfica.

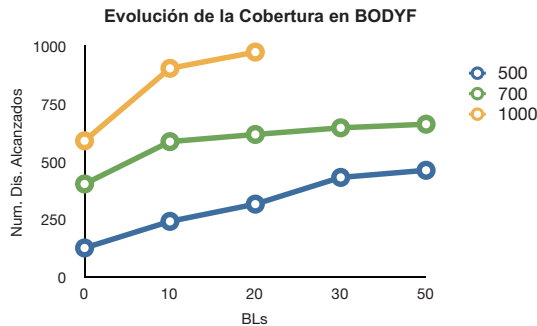


Fig. 8. Cobertura alcanzada en las distintas densidades.

En la Figura 8, se ve claramente como para la red de 500 dispositivos, utilizar más de 30 BLs no parece demasiado beneficioso. En las redes de 700 y 1000 se observa que el mayor escalón se produce en los 10 primeros BLs, tras ese punto la curva es muy suave, lo que significa que el incremento en la cobertura se va haciendo cada vez más pequeño.

VI. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se estudia el efecto de añadir enlaces de comunicación (BLs) entre dispositivos distantes en VANETs para conseguir incrementar la conectividad de la red. Para buscar la solución óptima se utiliza un algoritmo genético celular que trata de localizar de manera eficiente estos enlaces puente. Este estudio puede servir de referencia para evaluar la calidad de otros posibles algoritmos que sean capaces de resolver el problema de forma descentralizada. Además, da una idea del número de BLs necesarios para encontrar buenas soluciones.

Como resultado, se llega a la conclusión de que tan sólo con el uso de 10 BLs se permite duplicar la cobertura de los protocolos. El número mínimo de BLs utilizados en este trabajo es de 10, lo cual supone el 4% de los dispositivos en una red de 500 nodos, el 2,86% en la red de 700 y el 2% en la red de 1000. Aumentar el número de BLs en SF, conlleva un aumento considerable en la cobertura, pero en el caso de BODYF, la diferencia es bastante menor y, además, esta diferencia disminuye conforme aumentamos la densidad de la red. De nuestro estudio concluimos que BODYF se comporta de forma similar sobre soluciones no particionadas (independientemente de su calidad), e incluso con redes formadas por algunas particiones, por lo que no es necesario encontrar soluciones altamente precisas al problema.

Como trabajo futuro en esta línea de investigación, se planea diseñar y probar algún método que calcule de manera descentralizada la localización de los BLs que se van a utilizar y, además, se podrán utilizar los resultados obtenidos aquí para evaluar la calidad de dicha propuesta. Otro estudio que se pretende realizar es un análisis más profundo del intervalo entre 0 y 10 BLs para observar bien cómo se produce esta evolución de la cobertura obtenida.

REFERENCIAS

- [1] E. Alba y B. Dorronsoro. *Cellular Genetic Algorithms*. Operations Research/Computer Science Interfaces. Springer-Verlag Heidelberg, 2008.
- [2] E. Alba, B. Dorronsoro, F. Luna, A. J. Nebro, P. Bouvry, y L. Hogie. A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan manets. *Computer Communications*, 30(4):685–697, 2007.
- [3] T. Bäck, D. Fogel, y Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford Univ. Press, 1997.
- [4] CALM, Continuous communication for vehicles. <http://www.calm.hu/>, last accessed in may, 2008.
- [5] Car to car communication consortium. <http://www.car-to-car.org/>, last accessed in may, 2008.
- [6] C.A. Coello, G.B. Lamont y D.A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. 2a Edición. Springer-Verlag, 2007.
- [7] G. Danoy, E. Alba, P. Bouvry, y M.R. Brust. Optimal design of ad hoc injection networks by using genetic algorithms. En *Proc. of the conf. on genetic and evolutionary computation (GECCO)*, pages 2256–2256, 2007.
- [8] D. Gorgen, H. Frey, y C. Hiedels. JANE—the Java Ad-hoc Network Environment. En *Proceedings of the 40th Annual Simulation Symposium*, pages 163–176, 2007.
- [9] High Speed Downlink Packet Access (HSDPA). <http://www.telecomspace.com/latesttrends-hsdpa.html>.
- [10] J. Hubert, D. Weiler, y H. Brand. Umts, the mobile multimedia vision for imt-2000: A focus on standardization. *IEEE Communication Magazine*, 40:134–140, 2002.
- [11] S. Ni, Y. Tseng, Y. Chen, y J. Sheu. The broadcast storm problem in a mobile ad hoc network. En *Proc. of the ACM/IEEE Int. Conf. on Mobile Computing and Networking (MOBICOM)*, pages 151–162, 1999.
- [12] CARLINK project website. <http://carlink.lcc.uma.es>.
- [13] P. Ruiz, B. Dorronsoro, D. Khadraoui, P. Bouvry, y L. Tardón. BODYF – a parameterless broadcasting protocol over dynamic forest. En *International Conference on High Performance Computing and Simulation (HPCS 2008)*, pages 297–303, Nicosia, Cyprus, 2008.
- [14] A. S. Tanenbaum. *Redes de Computadoras*. Prentice Hall, 2003.
- [15] The Java Ad-hoc Network Environment (JANE). http://syssoft.uni-trier.de/jane/index.php/jane_basics.
- [16] Duncan J. Watts. *Small Worlds – The Dynamics of Networks between Order and Randomness*. Princeton University Press, Princeton, New Jersey, 1999.
- [17] B. Williams y T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. En *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 194–205, 2002.
- [18] A. Casteigts and S. Chamuette. Dynamicity Aware Graph Relabeling System - a local computation model to describe MANET algorithms. *Proceedings of the 17th IASTED International Conference on Parallel and Distributed Computing and System*, pages 198–204, 2005.
- [19] A. Casteigts. Model Driven Capabilities of the DA-GRS Model. *Intl. Conf. on Autonomic and Autonomous Systems (ICAS'06)*. San Francisco, USA, page 24, 2006.
- [20] S. J. Yang, H. K. Oh, y S. H. Park. Efficient multicast routing protocol for mobile hosts in IPv6 based networks. *Electronic Letters*, 38(16):936–938, 2002.
- [21] G. Danoy, P. Bouvry y L. Hogie. Coevolutionary Genetic Algorithms for Ad Hoc Injection Networks Design Optimization. *Proceedings of the IEEE Congress on Evolutionary Computation - CEC*, pages 4273–4280, 2007. Singapore