Contención limitada en respuesta al radiado mediante máscaras de elección de slot

Miguel Ortuño, Vicente Matellán, Clara Simón, José María Cañas
Departamento de Informática, Estadística y Telemática
Universidad Rey Juan Carlos - Móstoles - Madrid
{miguel.ortuno, vicente.matellan, clara.simon, josemaria.plaza}@urjc.es

Resumen

Conseguir que dispositivos autónomos se comuniquen de forma auto-organizada es un problema complejo que sólo se puede abordar descomponiéndolo en capas. El protocolo ADSR se ha diseñado para que dispositivos de este tipo, pero con capacidades limitadas (sensores, micro-robots, etc.) puedan crear redes ad-hoc inalámbricas. Para el funcionamiento de ADSR es necesaria la comunicación *broadcast* fiable en el nivel de enlace, lo que exije desarrollar el protocolo LLRB. Este protocolo a su vez necesita minimizar las colisiones de tramas en el radiado fiable, para lo que se ha diseñado el algoritmo que se describe en el presente artículo. Además, se aportan las simulaciones realizadas para validar su utilidad, así como un resumen del resto de características de ADSR y LLRB para poder situar el algorimo en su contexto.

1. Introducción

En informática, robótica y comunicaciones los equipos mejoran continuamente. Pero siempre habrá dispositivos sencillos, por diversos motivos. Se les puede exigir ser extremadamente fiables, baratos o desechables, como material docente, juguetes, colonias numerosas, microrobots o nanorobots. Puede que necesiten una vida útil muy larga con serias limitaciones de consumo energético que les obliguen a efectuar sólo las comunicaciones imprescindibles, por ejemplo en redes de sensores. Es muy probable que en el futuro se les encuentren aplicaciones que ahora no conocemos. Por todo ello nos atrevemos a afirmar que siempre tendrán su nicho las máquinas de recursos limitados.

En 2001 IBM desarrolla un concepto, la *Computación Autónoma*, (*Autonomic Computing*) [6], que es base de muchas iniciativas posteriores. Emplea la metáfora biológica del Sistema Nervioso Autónomo para expresar la necesidad de desarrollar sistemas informáticos capaces de auto-administrarse. Centrando esta idea en el ámbito de las comunicaciones, dentro del sexto programa marco de la Comisión Europea se acuña el término *Comunicaciones Autónomas* (*Autonomic Communications*) [3]. Es un paradigma donde aplicaciones y servicios no dependen de redes pre-existentes, sino que la red surge cuando la sitación y los servicios lo requieren, de forma autónoma, auto-organizada, distribuida, escalable e independiente de la tecnología. Podemos considerarlo un super-conjunto de las *Redes Ad-Hoc* [8].

El presente artículo se enmarca en una serie de trabajos que desarrollamos sobre encaminamiento en redes Ad-Hoc, basados en dos premisas fundamentales: Nos interesan las *máquinas de recursos limitados*, como sensores para aplicaciones domóticas y queremos *mantener el direccionamiento IP* dentro de la red Ad-Hoc. Mantener IP facilita enormemente la integración de la red Ad-Hoc en Internet: La red Ad-Hoc no necesita de Internet, ni de ninguna otra infraestructura, pero en caso de estar disponible el acceso, debe poder usarse. También facilita la reusabilidad y portabilidad del código. La contrapartida es que presenta dificul-

tades que exigen desarrollar nuevas técnicas, como el protocolo de red que hemos propuesto, ADSR [7], que exige un radiado (*broadcast*) fiable en el nivel de enlace. Para ofrecer esta prestación a ADSR, hemos diseñado el protocolo de enlace LLRB, presentamos aquí el segundo de los tres módulos en que se divide. En el apartado 2 indicamos los motivos que nos hacen desarrollar LLRB, en el apartado 3 describimos el acceso al medio de IEEE 802.11, en el que se basa LLRB. En el apartado 4 presentamos los aspectos básicos de su primer módulo, la principal aportación del presente artículo es el apartado 5, donde se describe el segundo módulo: *Contención limitada en respuesta al radiado mediante máscaras de elección de slot*, para terminar en el apartado 7 con las conclusiones.

2. Necesidad del radiado fiable en el nivel de enlace

Uno de los protocolos de encaminamiento en redes Ad-Hoc más extendidos es DSR (*Dynamic Source Routing*) [4]. Trabaja bajo demanda, cuando se necesita enviar un paquete a un destinatario y no se conoce una ruta válida, se hace una búsqueda por inundación [9]. La red se llena de paquetes de petición de ruta, cada paquete de petición registra en su interior el camino recorrido. Si una petición llega a su destino, puede hacer el camino inverso hasta el origen notificando la ruta. En lo sucesivo, *cada* paquete de datos lleva almacenada la dirección de *cada* nodo por el que debe pasar. Cada nodo a lo largo del camino es responsable de que el paquete de datos llegue hasta el siguente salto.

Si intentamos llevar el protocolo DSR sobre IPv4 a una arquitectura con un datagrama pequeño, digamos por ejemplo 256 bytes, tan solo las cabeceras ocuparían 88 bytes, un tercio del total disponible. Si usásemos IPv6, donde las direcciones son mayores, o arquitecturas con datagramas más pequeños, el problema haría inviable el uso de este protocolo.

2.1. Protocolo ADSR

Para solucionar este problema hemos propuesto el protocolo denominado *Abbreviated Dynamic Source Routing*, o ADSR [7]. Es una modificación drástica de DSR donde cada ruta no contiene la dirección de los nodos que la componen, sino un nuevo identificador o dirección abreviada que se construye a partir de la dirección original y que tendrá tamaño menor o igual. Esto supone romper la idea de que una dirección identifique de forma única a una estación: Podrá haber más de una máquina con la misma dirección abreviada, hecho al que denominamos *colisión*. Si $R = (D_1, D_2, ...D_n)$ es una ruta convencional como las que usa DSR, podremos abreviarla con cualquier función *Abb*() que genere nuevas direcciones que ocupen menos espacio, con tal de que la última dirección se mantenga. En IPv4 la función *Abb*() que elegimos devuelve una ruta formada por el último byte de cada nodo de la ruta original, excepto para la dirección del ultimo nodo, que no se modifica. El precio que se paga por el ahorro de espacio en las cabeceras son las colisiones.

2.2. Radiado fiable para ADSR: LLRB

En el protocolo DSR cuando un paquete con la ruta $R1 = (D_1, D_2, ..., D_i, D_{i+1}, ...D_n)$ llega a D_i, se reenvia a D_{i+1} , lo que supone un transmisión *unicast* ordinaria a una dirección conocida. Bajo el nivel de red en que trabaja DSR habrá un nivel de enlace con un esquema de direccionamiento distinto, pero entre la dirección de red y la de enlace habrá una correspondendia uno a uno que podrá resolverse con técnicas como ARP o similares.

Pero en ADSR dada una ruta
$r_1 = (d_1, d_2, ..., d_i, d_{i+1}, ..., d_n)$
el datagrama debe transmitirse desde d_i hasta d_{i+1}, donde d_{i+1} no identifica de forma única un nodo, por lo que este envío debe llegar a todas las máquinas cuya dirección abreviada coincida con d_{i+1}

Así, lo que para el nivel de red es un envío *unicast*, desde el punto de vista del nivel de enlace es una transmisión *multicast*. En otros ámbitos, el *multicast* más frecuente se produce

en el nivel de red, donde los nodos se añaden a un grupo con determinada dirección. Nótese que el *multicast* del que hablamos aquí es diferente, es un envío en nivel de enlace a todas las direcciones que cumplan determinada condición. Esto no está previsto por las arquitecturas convencionales e inevitablemente habrá que convertir el multicast en una las dos opciones disponibles:

- Varios *unicast*, lo que exige conocer las direcciones completas de todas las estaciones que deban recibir el envio.
- Un *broadcast*. Tras el cual, cada receptor, una vez que haya recibido el paquete lo descarta si su dirección abreviada no coincide con la de los destinatarios.

Para la primera opción habría que averiguar las direcciones de nivel de enlace de los nodos vecinos cuya dirección abreviada coincida con la del destinatario. Resultaría similar a un ARP convencional, con la salvedad de que la respuesta no sería una dirección sino un conjunto de direcciones. Para que el rendimiento del ARP sea bueno las cachés son fundamentales. Pero si una dirección de red se resuelve con un conjunto de direcciones de enlace, la caché no es aplicable puesto que pueden aparecer o desaparecer nodos de este conjunto.

Por ello, desestimamos el uso de los *unicast* y optamos por el radiado (*broadcast*). En una primera lectura el hacer todas las emisiones en modo radiado puede parecer cargar fuertemente el medio, pero no olvidemos que estamos en entorno inalámbrico, un *unicast* no es más que un radiado que es descartado por el nivel de enlace de todos los receptores excepto el del destinatario. Además, cada consulta ARP no deja de ser un radiado: Tendríamos tantos *unicast* como destinatarios, precedidos de un radiado de consulta ARP.

Recordemos además que en DSR cada nodo debe enviar un mensaje de error al origen en caso de que el paquete no haya podido alcanzar el siguiente salto. Pero en el radiado convencional las tramas perdidas no se detectan. Por todo ello, concluimos que para ADSR resulta imprescindible disponer de un radiado fiable en el nivel de enlace, para lo que diseñamos el protocolo LLRB (*Link Layer Reliable Broadcast*, radiado fiable en nivel de enlace). Este protocolo tiene tres módulos fundamentales, presentamos aquí el segundo de ellos: *Contención limitada en respuesta al radiado mediante máscaras de elección de slot*

Hemos decidido desarrollar LLRB a partir de IEEE 802.11, haciendo los cambios mínimos para satisfacer los requerimientos. Esto tiene la importante ventaja de que no es necesario partir de cero: Tanto para implementarlo en cualquier simulador de red como en hardware real, se puede tomar 802.11 como punto de partida y hacer las modificaciones que enumeramos. La contrapartida es que las máquinas de recursos limitados para las que el protocolo está diseñado no suelen contar con IEEE 802.11, que es un protocolo relativamente *caro*; lo llevan verdaderos ordenadores sin los límites de los que hablamos. Pero en esta fase del diseño nos resulta válido: Es bien conocido, sin duda el más extendido en la actualidad, sobre él está implementado DSR y por tanto ADSR. Describiremos a continuación el acceso al medio en IEEE 802.11, en el que se basa LLRB

3. Acceso al medio en IEEE-802.11

El comité IEEE 802.11 publicó en 1997 un conjunto de normas para redes de area local inalámbricas [2]. Contempla dos modos de operación: El modo *Ad-Hoc*, el esquema más sencillo donde todas las estaciones son iguales y el modo *Infrastructure*, que sigue el esquema cliente-servidor donde hay una estación principal, el *access point* (punto de acceso). Para redes *Ad-Hoc* se emplea el modo *Ad-Hoc*, pero aunque el término sea el mismo no debe confundirnos: Un aspecto fundamental de las redes *Ad-Hoc* es el encaminamiento, mientras que lo que IEEE-802.11 denomina modo *Ad-Hoc* es una tecnología estricamente de nivel de enlace.

El protocolo de acceso al medio empleado es DFWMAC, *Distributed Foundation Wireless Medium Access Control*. Proporciona un mecanismo de control distribuido (DCF, *Distri-*

buted Coordination Function, función de coordinación distribuida) usado en el modo *Ad-Hoc* y, opcionalmente, un control centralizado (PCF, *Point Coordination Function*, función de coordinación puntual) implementado sobre DCF. PFC es el mecanismo utilizado en el modo *Infrastructre*.

En IEEE 802.11, como en la mayoría de las tecnologías inalámbricas, no se permite emitir y recibir simultáneamente, lo que impide la detección de colisiones. Otros problemas bien conocidos son el del *Nodo Oculto* y *Nodo Expuesto*. Para evitar estos inconvenientes, IEEE 802.11 integra el mecanismo RTS/CTS desarrollado en MACA (*MultiAccess Collision Advoidance* [5]) (figura 1), combinado con un esquema de prioridades de las tramas. Cuando un nodo desea emitir un dato, previamente envía al destinatario una trama especial, breve, denominada RTS (*Request To Send*, petición de envio). El destinatario responde con un mensaje CTS (*Clear to Send*). Las estaciones próximas sabrán que el medio estará ocupado el tiempo necesario para intercambiar una trama de datos. No importa si sólo han podido oir el RTS, sólo el CTS o ambos; en todo caso considerarán el medio ocupado. A este mecanismo se le llama *detección de portadora virtual*: Una variable denominada NAV (*Network Allocation Vector* Vector de Asignación de red) almacenará la hora hasta la que se considera el medio ocupado. Si la hora actual es anterior a NAV, el medio no está libre. Tras el intercambio con éxito del RTS y CTS, se envía la trama de datos, que se asiente con una trama ACK (*Acknowledgment*). Esta secuencia RTS-CTS-DATO-ACK se aplica a los paquetes de datos *unicast*, en IEEE-802.11 los paquetes de radiado no son asentidos.

Ademas de todo esto, se establecen prioriades para las tramas: Cuando una estación desea transmitir, sondea el medio. Si en el primer intento detecta que está libre, y que permanece libre durante un tiempo IFS (*InterFrame Space*, espacio entre tramas), la estación empieza a emitir inmediatamente. En cualquier otro caso, sigue reintentando hasta percibir el medio libre durante un tiempo IFS. Entonces espera, además, una *ventana de contención*, y

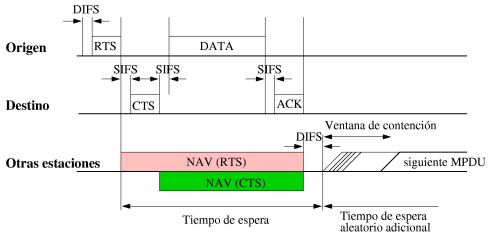

Figura 1: Mecanismo RTS CTS ACK

si el medio aún sigue libre, emite.

Este tiempo IFS no es fijo, puede tener 4 valores diferentes, que permiten priorizar las tramas. Ordenados en orden creciente son: SIFS (*Short InterFrame space*, espacio corto entre tramas), el tiempo que se espera para enviar un CTS tras oir un RTS, también para enviar un ACK después de recibir un dato. PIFS (*PCF InterFrame space*, espacio entre tramas de la función de coordinación puntual), el tiempo que espera un *access point* antes de enviar una trama de control. DIFS (*DCF InterFrame space*, espacio entre tramas de la función de coordinación distribuida), el tiempo que se espera antes de que responda una estación en modo *ad-hoc*. EIFS (*Extended InterFrame space*, espacio extendido entre tramas), el tiempo esperado antes de enviar notificaciones de error en una trama.

4. Extensión del mecanismo RTS/CTS/ACK para múltiples destinatarios

Para que LLRB permita ofrecer radiado fiable, se define la dirección ff:ff:ff:ff:ff:fe. Cualquier nodo que reciba una trama con este destinatario, debe aceptarla y enviar confirmación. Las tramas de radiado fiable coexisten con las de radiado convencional no asentido, de dirección ff:ff:ff:ff:ff:ff. Las tramas ACK de LLRB indican explícitamente la dirección de la estación que las transmite (en IEEE 802.11 un nodo que emite esta trama queda identificada de forma implícita, sólo una estación era destinataria del dato). También se identifica explicitamente el emisor del CTS.

El mecanismo RTS/CTS/ACK convencional se caracteriza por ser un diálogo entre dos estaciones concretas, mientras que en un radiado fiable, todas las estaciones que escuchen el RTS deben, en principio, contestar CTS y todas las que reciban el dato deben enviar un ACK. Esto plantea el problema clásico de varias estaciones compitiendo por el medio. La forma más sencilla de resolverlo es mediante la conocida técnica del *Aloha* ranurado: Definiremos un nuevo espacio entre tramas: LIFS (*LLRB InterFrame space*, espacio entre tramas del radiado fiable en nivel de enlace). Lo representamos en la figura 2. Será superior a SIFS e inferior a DIFS. LIFS se segmenta en ranuras (*slots*) de forma completamente análoga a la ventana de contención, cada nodo elige aleatoriamente una ranura, y si en ese momento el medio está libre, emite. La longitud de cada ranura será el tiempo necesario para transmitir un CTS o un ACK, más un SIFS.

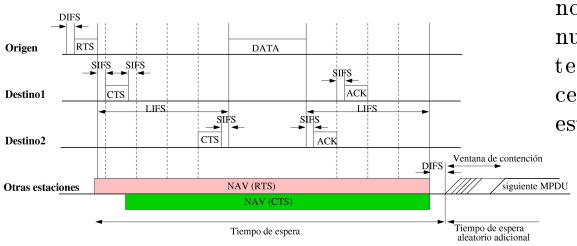

Figura 2: Mecanismo RTS CTS ACK en LLRB

En MACA convencional, cuando un nodo percibe un RTS que no va dirigido a él, o cuando percibe un CTS, sabe que alguno de sus vecinos están enviándose datos, por tanto actualiza la portadora virtual y considera el medio ocupado el tiempo suficiente para que la transmisión finalice. En LLRB cuando un nodo envía un RTS a la dirección de radiado fiable, el primer vecino que gane la contienda enviará un CTS. Pero los vecinos siguientes también deben enviar su CTS, para este fin deben ignorar que la portadora virtual considera el medio ocupado. El emisor dará oportunidad a que todos los destinatarios envien su CTS, transcurrido este tiempo, enviará la trama de datos. Es posible que alguna estación reciba este dato antes de poder enviar su propio CTS. En ese caso, aceptará la trama de datos y desistirá en el envío del CTS:

4.1. Análisis de *Aloha* ranurado

El acceso al medio incluido en el primer módulo de LLRB emplea *aloha* ranurado: Después del envío de un RTS o una trama de datos, las estaciones vecinas responden CTS o ACK en una ranura aleatoria. Analizaremos a continuación las probabilidades de colisión con este acceso al medio. Aplicando el modelo de *la paradoja del cumpleaños* [1], siendo s el número de ranuras y n el número de vecinos compitiendo por el medio, la probabilidad de que dos de ellos usen el mismo *slot*, esto es, $p(c) = p(\text{"}al\ menos\ una\ colision\text{"})$ es superior a 0,5 con $n \geq \frac{1+\sqrt{1+(8 \ln 2)s}}{2}$

Según el modelo del *coleccionista de cromos* [1], cabe esperar que el número medio de vecinos necesarios para que no quede ninguna ranura libre es $s \ln s$. Si bien mucho antes de este valor el rendimiento sería posiblemente inaceptable. En el cuadro 1 vemos el resultado de estas expresiones para algunos ejemplos.

s	$p(c) > 0{,}5$	$s \ln s$
4	3	6
8	4	17
16	5	44
32	7	111
64	10	266
128	14	621
365	23	2153

Cuadro 1: Limitaciones de *aloha* ranurado

5. Contención limitada en respuesta al radiado mediante máscaras de elección de slot

El segundo módulo de LLRB (*Contención limitada en respuesta al radiado mediante máscaras de elección de slot*) permite elegir de forma determinista la ranura en que contestan las estaciones, mejorando el rendimiento frente a *aloha* ranurado. Expondremos la idea básica con un caso sencillo:

El nodo que transmite no sabe exactamente cuáles son sus vecinos, pero sí tine una estimación: conoce las direcciones del vecindario que tuvo en el tiempo precedente. Ignoraremos ahora las distorsiones que provoque el error en la estimación del vecindario. Supongamos que el nodo A tiene 3 vecinos, B, C y D cuyas direcciones son las indicadas en el cuadro 2. Supongamos que el espacio LIFS está divido en 4 ranuras. Sea n el número de vecinos y s el número ranuras. Si A le comunica a cada vecino qué ranura usar, no habrá colisiones. Pero es fundamental que esa comunicación tenga un coste mínimo. Esto puede hacerse de forma eficaz incluyendo en el RTS o en el ACK el mensaje *'usad los bits 0 y 2 de vuestra dirección para elegir un slot'*. O lo que es lo mismo, *'usad la máscara 00000101'*. De esta manera, las estaciones B, C y D extraerían los bits de su dirección que se correnspondan con bits altos en la máscara, obteniéndose respectivamente *11*, *00* y *01* según muestra el cuadro 2. Esto indica el cuarto, primer y segundo slot respectivamente, con lo que en este ejemplo no se producen colisiones.

	7	6	5	4	3	2	1	0
Mask	0	0	0	0	0	1	0	1
B	0	0	0	0	1	1	1	1
C	0	0	0	0	1	0	1	0
D	0	0	0	0	0	0	1	1

Cuadro 2: Ejemplo de máscara de elección de slot

Si $n > s$ será inevitable que se produzcan colisiones. También es posible que las direcciones de los nodos sean especialmente desfavorables, y aún con $n \leq s$ no exista una máscara que asigne a cada vecino una ranura diferente. Si bien A sabe que se producirá una colisión desde el momento en que genera la máscara. Se trata por tanto de diseñar un algoritmo eficiente que elija las máscaras de forma que no se produzcan colisiones. Y si esto es imposible, que se repartan equitativamente las colisiones de tal forma que con un número mínimo de reintentos, empleando máscaras diferentes en cada intento, todos los vecinos hayan encontrado al menos una vez una ranura libre.

5.1. Algorimo de elección máscara

En primer lugar se establece s, el número de ranuras que se emplearán. Un número alto reducirá las colisiones y aumentará la fiabilidad del protocolo, con el inconveniente de aumentar el tiempo entre tramas y disminuir el ancho de banda disponible. Un valor muy bajo de s provocará que apenas lleguen CTS ni ACK, provocando prácticamente el colapso del sistema. El valor de s determina b, el número bits altos en la máscara, donde $2^b = s$. Por tanto s habrá de ser potencia de 2. Si el emisor decide variar s, basta con que empiece a usar máscaras con el número adecuado de bits altos. Los receptores contarán el número de unos en la máscara y actuarán en consecuencia. También los nodos vecinos, que actualizarán el valor de la portadora virtual de acuerdo con esto.

La máscara tiene un tamaño de 8 bits, y se aplica siempre sobre el último byte de la dirección de cada nodo. Esto garantiza que ni la elección ni la transmisión al destinatario sea muy costosa. El emisor considerará todas las máscaras posibles, y entre ellas seleccionará las que considere *máscaras buenas*. Este conjunto de máscaras será válido mientras no cambie el vecindario. En cada retransmisión de una trama se emplea una máscara diferente, siempre dentro de las *buenas*. Si despues de utilizar todas las *máscaras buenas* no han llegado todos los CTS y ACK, no tiene sentido reusarlas de nuevo, generarán colisiones en los mismos sítios. Entonces se enviará la *máscara nula*, sin ningún bit alto. Significa que los destinatarios deben elegir un ranura aleatoria. En el caso de que se detecte que dos (o más) vecinos tengan igual el último *byte*, tras enviar una máscara de las *buenas* se enviará la *máscara nula*, puesto que ninguna máscara ordinaria permitirá que estos vecinos usen diferentes ranuras.

Vemos ahora cómo elegir las *máscaras buenas*: El emisor divide el número de vecinos entre el número de ranuras para calcular el *reparto óptimo*: el número máximo de vecinos que deberían ocupar cada slot en una distribución homogénea. Se consideran todas las máscaras con b bits altos, y se desechan aquellas por encima del reparto óptimo. Todas estas operaciones con máscaras son muy *baratas* en cual-

quier arquitectura. Respecto al número m de máscaras con b bits altos a explorar, se obtiene de combinar b en 8 posiciones diferentes, donde $m = \left(\begin{array}{c} 8 \\ b \end{array} \right) = \frac{8!}{b!\,(8-b)!}$. Los valores posibles de m se muestran en el cuadro 3, se observa que son valores razonablemente bajos.

b	1	2	3	4	5	6	7
m	8	28	56	70	56	28	8

Cuadro 3: Número de máscaras posibles

Aún aplicando este algoritmo, seguirá habiendo colisiones y por tanto pérdidas. Las colisiones podrán ser diferentes a lo supuesto por el emisor si la estimación del vecindario no es precisa. Pero no debemos olvidar que estamos hablando de pérdidas de tramas CTS y ACK, no de datos: Sólo una de las estaciones es la que emite datos, ninguna otra estación vecina intentará emitir una trama de datos si ha percibido un RTS o un CTS.

Además, en cada envío hay un único RTS, que provocará la respuesta de múltiples CTS. A esto le seguirá una única trama de datos y múltiples ACK. La pérdida de un CTS es irrelevante si ha llegado un ACK de ese nodo. (El objetivo fundamental del CTS es que nodos vecinos consideren el medio ocupado, pero para esto basta un CTS). La pérdida de uno de los ACK supone un inconveniente mayor, pero si el emisor ha percibido el CTS, tiene constancia de que el nodo sigue siendo su vecino, y por tanto, puede haber oido la trama de datos. Teniendo esto en cuenta, una vez fijado el número s de ranuras, estas pueden situarse dentro de un periodo LIFS, o repartirse entre el LIFS anterior y el posterior a la trama de datos. De este modo, no todas las estaciones intentarían enviar siempre tanto CTS como ACK, conformándose con una de las dos tramas. También pueden repartirse estas ranuras entre varias secuencias RTS-CTS-DATO-ACK. Así, no todas las estaciones contestarán todas las tramas, dándose por supuesto que se harán varios intentos. Esto será ventajoso con niveles de carga elevados.

6. Experimentación

Hemos implementado una simulación del algoritmo en C++, disponible en `http://193.147.71.64/~mortuno/ucami_simuls.tgz`
Los resultados son bastante satisfactorios: la figura 3 representa el número de tramas CTS y ACK entregadas en el primer intento en función del número de vecinos, para 8, 16, 32 y 64 ranuras. La figura 4 representa el número de envios necesarios hasta que todas las tramas son entregadas. Usa una escala logarítmica, observamos como en la inmensa mayoria de los casos se pueden hacer llegar todas las tramas con un máximo de 6 intentos. Estos resultados permiten validar la utilidad del algoritmo, así como elegir el número adecuado de ranuras en función de la carga en la red.

Podemos ilustrar el escaso consumo de recursos del algoritmo indicando que todos los cálculos realizados con cada combinación de vecinos, no necesitaron más de 700 microsegundos de media para simularse en un portátil con CPU Pentium M a 1700Mhz (3300 BogoMips).

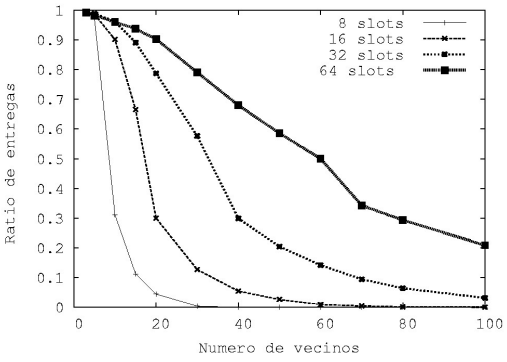

Figura 3: Ratio de paquetes entregados en el primer envío

7. Conclusiones

Cuando los recursos son limitados y las redes se forman y destruyen sobre la marcha y los recursos son limitados, a cada nodo le re-

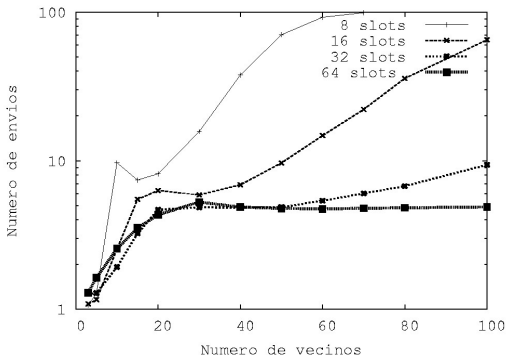

Figura 4: Número de envíos necesarios hasta asentir todas las tramas

sulta costoso conocer la identidad de sus fugaces vecinos. Pero es necesario para garantizar la comunicación con todos ellos, especialmente si se desea mantener las direcciones IP, cuyo tamaño en sí mismo puede suponer un problema. Es necesario un radiado fiable en nivel de enlace que sea *barato*. Para ello desarrollamos el protocolo de enlace LLRB, que permite que varias estaciones compartan el medio para contestar CTS y ACK a un único emisor de datos, que puede coordiarlas. El espacio de tiempo entre las tramas se ranura, y el emisor indica a cada nodo cuándo puede contestar. Hemos propuesto un método en el que el emisor, a partir de una estimación de su vecindario genera una máscara de un byte. Cada destinatario aplica la máscara a su propia dirección para obtener la ranura de tiempo en la que enviar su respuesta. Esto permite dispersar las respuestas para evitar colisiones, dentro de ciertos límites que hemos caracterizado. En cuanto al trabajo futuro, estamos desarrollando el tercer y último módulo de LLRB, así como la integración entre los tres módulos y ADSR en un simulador más completo.

Agradecimientos

Los autores agradecen los comentarios y sugerencias de Antonio Fernández Anta.

Referencias

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition.* The Massachusetts Institute of Technology, 2001.

[2] B. P. Crow. IEEE 802.11 wireless local area networks. *IEEE Communications Magazine*, september 1997.

[3] European Commmission. EU IST FET, Situated and Autonomic Communications (COMS) - Communication Paradigms for 2020. http://www.cordis.lu/ist/fet/comms.htm, July 2003.

[4] D. Johnson, D. Maltz, and J. Broch. *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[5] P. Karn. MACA - a new channel access method for packet radio. In *Amateur Radio 9th Computer Networking Conference*, pages 134–140, 1990.

[6] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *IEEE Computer Magazine*, 36(1):41–50, 2003.

[7] M. A. Ortuño, V. Matellán, L. Rodero, and G. Robles. Abbreviated dynamic source routing: Source routing with non-unique network identifiers. In *Proceedings of WONS 2005. Second Annual Conference on Wireless On-demand Network Systems and Services. IEEE Computer Society*, pages 76–82, 2005.

[8] C. E. Perkins. *Ad Hoc Networking.* Addison-Wesley, 2001.

[9] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. IP flooding in ad hoc mobile networks. www.ietf.org/proceedings/01dec/I-D/draft-ietf-manet-bcast-00.txt, Mar. 2001. IETF Internet Draft.