

EXTENSIÓN DEL MECANISMO RTS/CTS/ACK PARA MÚLTIPLES DESTINATARIOS

Miguel Ortuño, Vicente Matellán, José María Cañas, Carlos Agüero
ESCET – Universidad Rey Juan Carlos
Móstoles, Madrid
{mortuno,vmo,jmplaza,caguero}@gsyc.escet.urjc.es

Abstract

Using IP addresses in limited devices provides many benefits, but the big size of the addresses becomes a problem when resources are bounded. In order to send data to all its neighbors in an Ad-Hoc Network, a node must know all their identities, which can have a severe impact on available resources. Then, a link layer reliable broadcast emerges as a necessary feature. In this paper we present a protocolo named LLRB (*Link Layer Reliable Broadcast*) that extends the RTS/CTS/ACK mechanism to be used when the data is sent not only to a single node but to the whole neighborhood.

PALABRAS CLAVES

Redes Ad-Hoc, Nivel de Enlace, MACA, Radiado Fiable

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. En 2001 IBM desarrolla un concepto, la *Computación Autónoma*, (*Autonomic Computing*) [7], 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 situación 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* [10].

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* y queremos *mantener el direccionamiento IP* dentro de la red Ad-Hoc.

Apreciamos una tendencia clara a conectar a Internet todo tipo de equipos, grandes o pequeños. Aunque se pueda comparar el número de direcciones disponibles en IPv6 con el número de moléculas en la tierra, muchas voces afirman que no tiene sentido integrar todos los dispositivos mundiales en una única *red de redes* mastodóntica. Al contrario, se debe potenciar la autonomía de las redes. Esto no quiere decir que las redes Ad-Hoc estén aisladas de Internet: podrán conectarse a ella en las ocasiones en que resulte necesario y haya una pasarela accesible.

El objetivo fundamental de las redes Ad-Hoc es minimizar su dependencia de infraestructuras previas. Aunque si hay alguna disponible, por ejemplo un *gateway* que de acceso a Internet, debe poder usarse. Normalmente las redes Ad-Hoc emplean esquemas de direccionamiento específicos para los dispositivos pequeños en los que IP no resulta apropiado. El *gateway* deberá realizar la traducción entre ambas direcciones. Si el nodo se mueve y usa un *gateway* diferente, perderá su dirección IP (a menos que todos los *gateways* se coordinen entre sí). Debe haber también una entidad encargada de proporcionar las direcciones no-IP, garantizando su unicidad.

Este es el enfoque predominante, pero en nuestro trabajo nos planteamos una aproximación diferente que nos parece ventajosa: Para facilitar su conexión a Internet, las redes Ad-Hoc deben mantener en todo momento el

direccionamiento IP. Eso elimina la necesidad de traducir direcciones. O dicho de otro modo, para proporcionar conectividad con Internet bastará un *router*, no siendo necesario un *gateway*. Tampoco será necesario contar con una entidad propia que asigne direcciones únicas (basta el organismo correspondiente de Internet). Así mismo, mantener IP permite reutilizar código ya existente y hacer aplicaciones más portables. La contrapartida es que presenta dificultades que exigen desarrollar nuevas técnicas.

Mantener una estructura como IP que no está pensada para redes Ad-Hoc puede parecer un lastre, pero a nadie se le escapa que la compatibilidad con la tecnología precedente es un factor determinante.

Para solventar los problemas que plantea el uso de IP en máquinas de recursos limitados, hemos propuesto el protocolo de red ADSR, cuyo rendimiento es muy limitado si no dispone de un radiado (*broadcast*) fiable en el nivel de enlace. Para ofrecer esta prestación a ADSR, hemos diseñado el protocolo de enlace LLRB (*Link Layer Reliable Broadcast*, radiado fiable en nivel de enlace), presentamos aquí el primero de los tres módulos en que se divide. En el apartado 2 indicamos los motivos que nos hacen desarrollar el protocolo, en el apartado 3 describimos el acceso al medio de IEEE 802.11, en el que se basa LLRB. En el apartado 4 exponemos con detalle el primer módulo de LLRB: *Extensión del mecanismo RTS/CTS/ACK para múltiples destinatarios*, del que tenemos disponible una implementación en el simulador ns-2 [4]. Mostramos algunos resultados preliminares en el apartado 5, para terminar en el apartado 6 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*) [5]. Trabaja bajo demanda y hace encaminamiento en origen: *cada* paquete de datos lleva en su cabecera la dirección de *cada* nodo por el que debe pasar. Si intentamos llevar el protocolo DSR sobre IPv4 a una arquitectura con un datagrama pequeño, digamos por ejemplo 256 bytes, solo las cabeceras ocuparían un tercio del total disponible. Las direcciones de todos los nodos de la ruta consumen la mayor parte de estas cabeceras. Si usásemos IPv6, donde las direcciones son mayores, o arquitecturas con datagramas más pequeños, el problema se agravaría, resultando completamente inviable el uso de este protocolo.

2.1 PROTOCOLO ADSR

Para entornos donde DSR no es aplicable, hemos propuesto el protocolo denominado *Abbreviated Dynamic Source Routing*, o ADSR [8] [9]. Es una modificación drástica de DSR para máquinas de recursos limitados, 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 $R=(D_1, D_2, \dots, D_n)$ es una ruta convencional como las que usa DSR, donde D_i es una dirección IP, podremos abreviarla con cualquier función $Abb()$ que genere nuevas direcciones que ocupen menos espacio, con tal de que la última dirección, D_n , 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 último nodo, que no se modifica. El precio que se paga por este 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 $R_1=(D_1, D_2, \dots, D_i, D_{i+1}, \dots, D_n)$ llega a D_i , se reenvía a D_{i+1} , lo que supone una 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 correspondencia uno a uno que podrá resolverse con técnicas como ARP o similares. Pero en ADSR dada una ruta $r_2=(d_1, d_2, \dots, d_i, d_{i+1}, \dots, d_n)$ 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*. Nótese que el *multicast* del que hablamos aquí no es en nivel de red, 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 envío.
- 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 MAC 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. Un consulta ARP convencional solo tiene una respuesta posible, siempre la misma. Pero si una dirección de red se resuelve con un conjunto de direcciones de enlace sería muy costoso mantener la coherencia de la caché: la respuesta al ARP cambia si el vecindario cambia.

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 que ADSR pueda notificar errores al origen 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:

1. Extensión del mecanismo RTS/CTS/ACK para múltiples destinatarios
2. Limitación de las colisiones en respuesta al radiado
3. Algoritmo ligero de estimación de vecindario

En el presente trabajo trataremos exclusivamente el primero, del que tenemos una implementación¹. A pesar de no tratarse del protocolo completo, lo consideramos de interés. Hemos decidido desarrollar LLRB a partir de IEEE 802.11, haciendo los cambios mínimos para satisfacer los requerimientos. Esto tiene la importante ventaja de permite 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.

Antes de describir la extensión del mecanismo RTS/CTS/ACK del protocolo LLRB, en el siguiente apartado describimos el acceso al medio en IEEE-802.11

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].

En este protocolo, 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 el del *nodo expuesto* [6]. Para evitar estos inconvenientes, IEEE 802.11 integra el mecanismo RTS/CTS desarrollado en MACA (*MultiAccess Collision Avoidance* [6]), combinado con un esquema de prioridades de las tramas; si bien no elimina los problemas por completo, obtiene una mejora muy importante [11]. Cuando un nodo desea emitir un dato, previamente envía al destinatario una trama especial, breve, denominada RTS (*Request To Send*, petición de envío). 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 oír 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

¹Aun no consideramos que esté lo suficiente madura como para ser publicado, aunque la versión actual puede solicitarse a los autores

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.

Además de todo esto, se establecen prioridades 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 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). Es el tiempo que se espera para enviar un CTS tras oír 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). Es el tiempo que se espera antes de que responda una estación en modo *ad-hoc*.
- EIFS (*Extended InterFrame space*, espacio extendido entre tramas). Es el tiempo esperado antes de enviar notificaciones de error en una trama.

4 EXTENSIÓN DEL MECANISMO RTS/CTS/ACK PARA MÚLTIPLES DESTINATARIOS

Una vez descritas la motivación y las bases, detallamos el primer módulo del protocolo LLRB y principal aportación del presente trabajo.

4.1 RADIADO FIABLE Y RADIADO ORDINARIO

La dirección de radiado clásico se representa en hexadecimal como ff:ff:ff:ff:ff:ff. Para el radiado fiable del protocolo LLRB definiremos la dirección ff:ff:ff:ff:ff:fe. Así, en IEEE-802.11, cuando un nodo recibe una trama comprueba la dirección del destinatario, si no coincide con la suya, la descarta. Mientras que en LLRB, la trama se acepta si coincide con la suya o con la dirección de radiado fiable.

Las tramas de radiado convencional de IEEE 802.11 son necesariamente pequeñas: Al enviarse sin RTS/CTS la probabilidad de colisión aumenta. Las tramas de radiado fiable irán protegidas por el mecanismo de la portadora virtual, con lo que no tendrán esta limitación y pueden tener el mismo tamaño de una trama *unicast*.

En LLRB no siempre necesitaremos un radiado fiable. Por ejemplo en ADSR el descubrimiento de ruta se realiza con una inundación controlada de peticiones de ruta en paquetes de radiado convencionales. Por tanto, ambos tipos de radiado deberán coexistir.

4.2 LIFS: LLRB INTERFRAME SPACE

El mecanismo RTS/CTS/ACK convencional se caracteriza por ser un diálogo entre dos estaciones concretas: Sólo el nodo destinatario del RTS responderá CTS, sólo el receptor de una trama de datos contesta ACK. 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.

Inicialmente lo resolveremos con la conocida técnica del *Aloha* ranurado². Definiremos un nuevo espacio entre tramas: LIFS (*LLRB InterFrame space*, espacio entre tramas del radiado fiable en nivel de enlace). Será superior a SIFS e inferior a DIFS. Al estar nuestro trabajo orientado a redes *ad-hoc*, no necesitamos considerar su relación con PIFS, como sucede con todo lo relativo al modo *infrastructure*.

LIFS se segmenta en ranuras de forma completamente análoga a la ventana de contención, cada nodo elige

² El segundo módulo de LLRB mejora este mecanismo, mediante elección determinista del slot

aleatoriamente un segmento, 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.

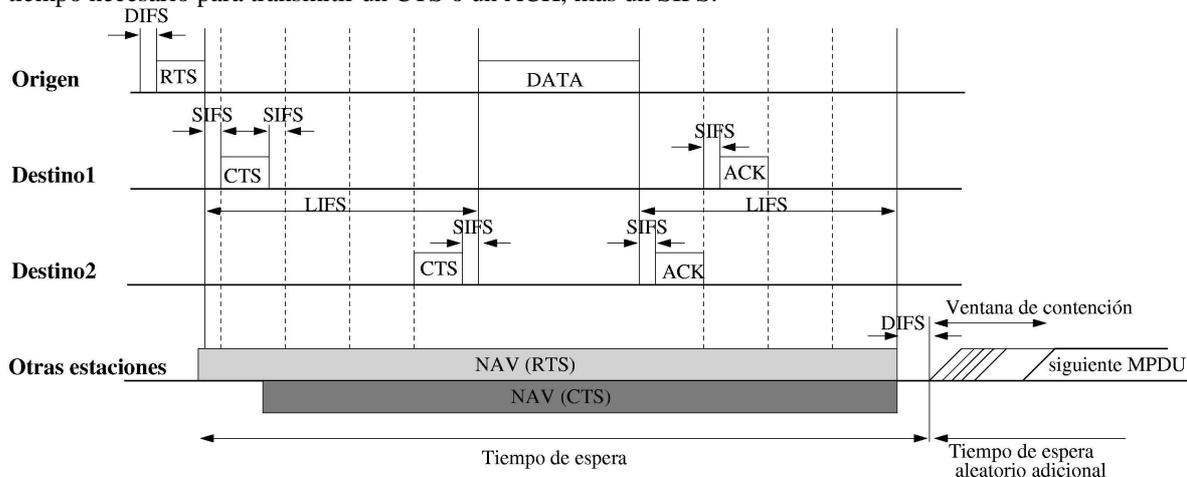


Figura 1. Extensión del mecanismos RTS/CTS/ACK para múltiples destinatarios

4.3 IDENTIFICACIÓN DE CTSS Y ACKS

En el MACA convencional, una estación que envía un ACK queda identificada de forma implícita: Cuando una estación recibe un ACK sabe que sólo puede venir del nodo al que acaba de enviarle un dato. Un asentimiento escuchado en cualquier otro momento se ignora. Un aspecto fundamental de LLRB es controlar qué nodos están confirmando el paquete, es necesario identificarlos explícitamente. Por tanto modificamos la trama ACK para añadir la dirección de origen.

Con los CTS sucede lo mismo: cualquier nodo oyendo un RTS puede contestar CTS, así que en la trama añadimos la dirección del que está respondiendo.

4.4 MULTIPLICIDAD DE LOS RTSS

Siguiendo el estándar IEEE 802.11, un RTS reserva el medio y sólo una estación contesta CTS, con lo que la probabilidad de colisión es relativamente baja. Si durante un intervalo de tiempo $CTSTimeout$ no llega respuesta, se reenvía el RTS. El número de reintentos se controla en la variable *short retry count*, cuando esta alcanza el valor límite *ShortRetryLimit*, desiste. Los valores concretos de todas estas variables no están definidos en el estándar, eligiendo cada implementación los que considere adecuados. Por ejemplo el simulador ns-2 toma el tiempo de transmisión del RTS más el del CTS más dos veces el máximo retardo de propagación empleando DSSS (la modulación de 802.11) y un SIFS.

En LLRB, debemos adaptar este *timeout* teniendo en cuenta que hay que esperar un tiempo LIFS para dar la oportunidad de contestar a todos los receptores. Al valor *ShortRetryLimit* se le aplica un incremento $\Delta_{ShortRetryLimit}$, ya que en LLRB hay muchos más CTSS y ACKs que en IEEE 802.11, por tanto aumenta la probabilidad de que el medio esté ocupado. Además, si el destinatario de la trama de datos ha recibido un RTS pero no ha podido responder CTS por estar el medio ocupado, ignorará los sucesivos RTS que pueda recibir.

4.5 MULTIPLICIDAD DE LOS CTSS

Según el estándar 802.11 a un CTS debería seguirle una trama de datos, pero ni se hacen reenvíos ni se controla ningún *timeout*. Las implementaciones pueden controlar que este tiempo ha vencido, pero la única acción es pasar a un estado inactivo.

Para describir las peculiaridades de los CTSS en nuestro protocolo, supongamos 3 nodos que cuentan con los mecanismos LLRB descritos hasta ahora, en una disposición como la de la figura siguiente. (A y C se ven entre sí)

```

A           B           C
          <--- RTS --->
          CTS--->
          <--- DATA --->
          ACK--->

```

El nodo B desea enviar una trama de datos con radiado fiable, deben recibirla sus vecinos A y C. B envía una trama RTS, A y C la reciben, A gana la contienda a C y escoge una ranura dentro dentro del tiempo LIFS, contestando el CTS correspondiente.

Cuando B recibe el primer CTS, espera el valor máximo de LIFS para dar oportunidad de que se ocupen todos los slots, y emite la trama de datos. Obviamente, B no puede estar seguro de que todos los nodos hayan contestado CTS: no tiene forma de saber cuántos son *todos*, es posible que haya destinatarios que no hayan enviado su CTS.

C oye el CTS e incrementa su NAV, considerando el medio ocupado el tiempo necesario para el envío y confirmación del dato, por tanto no envía su propio CTS.

El dato será oído por A, que lo recibe y procesa correctamente, contestando ACK. También C lo recibe, pero como aún no ha enviado su CTS, no espera recibir un dato, y lo descarta, con lo que no hemos conseguido el radiado buscado. Esto podría solucionarse de diferentes maneras:

1. Cuando C percibe el CTS de A, considera el medio ocupado y no envía su propio CTS, pero da por buena la trama de datos cuando la recibe: Acepta un dato tanto después de recibir un RTS como después de enviar un CTS. Esto ahorra energía y carga menos el medio, el inconveniente es que por no enviar CTS, un cuarto nodo D próximo a C pero distante de A considere el medio libre y provoque una colisión. Además, no enviar un CTS supone dejar escapar una oportunidad de notificarle a B la presencia de C, lo que será conveniente para calcular el vecindario estimado, ya que el ACK que envíe C posteriormente podría perderse en una colisión.
2. C percibe el CTS de A, pero no actualiza la portadora virtual y considera el medio libre para enviar su propio CTS, tras lo cual acepta la trama de datos. Esto presenta el inconveniente de que tal vez no llegue a enviar el CTS por no quedar *slots* libres, y que por tanto aunque reciba la trama de datos, la considere fuera de secuencia y la ignore. Además, C está ocupando un *slot* que podría necesitar un cuarto nodo D que no haya percibido el CTS de A.
3. La solución por la que optamos es una combinación de las dos anteriores: Cuando C recibe el RTS, aunque perciba el CTS de A no considera el medio ocupado e intenta enviar su propio CTS. Pero si recibe la trama de datos antes de haberlo conseguido, la acepta en todo caso.

Nuestra solución también tiene en cuenta que la potencia con la que la señal de A llega a C puede estar o bien por encima del umbral que permite leer la trama, o bien por debajo. En previsión de este último caso, se usa un flag *last_rx_was_rts* (la última trama recibida fue un RTS). Con este flag activo, se supone que cualquier señal recibida, aunque no pueda leerse, es un CTS de un vecino y no debe actualizarse la portadora virtual.

4.6 CÁLCULO REDUNDANTE DEL NAV

En 802.11 la recepción de un RTS o de un CTS pone en la variable NAV la hora a la que se espera que ya se haya enviado y confirmado el paquete, y por tanto se puede considerar el medio libre. Cabe la posibilidad de que RTS y CTS se hayan perdido, así que como medida de seguridad adicional, cuando se oye una trama de datos destinada a otro nodo, se calcula el tiempo necesario para recibirlo por completo y asentirlo, y si NAV no se hubiera actualizado, se actualiza.

Construyendo LLRB como modificación de 802.11, no se nos debe escapar que esta actualización del NAV sólo debe hacerse si la trama no está dirigida ni a nuestra dirección, ni a la dirección de radiado fiable.

5 RESULTADOS PRELIMINARES

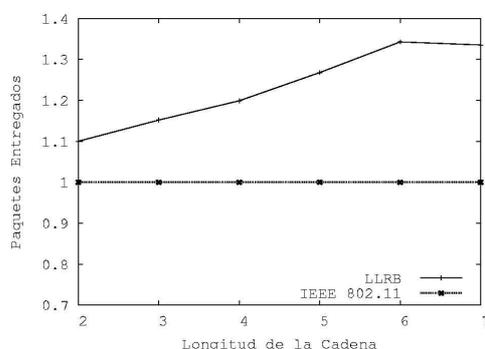


Figura 2. Ratio de paquetes de red entregados

Disponemos de una implementación del primer módulo del protocolo LLRB. Contamos por tanto con una versión modificada de IEEE 802.11 que envía tramas de datos a una dirección de radiado fiable, los nodos vecinos las reciben y las confirman con tramas ACK, donde los CTS y los ACK comparten el medio por *aloha* ranurado. No hay estimación de vecindario ni se envían errores al nivel de red. El simulador empleado es ns-2 [4].

A modo de ejemplo, colocamos una cadena de entre 2 y 7 nodos, de tal forma que todas las estaciones (excepto la primera y la última) verán 2 nodos: el anterior y el posterior. Por encima del nivel de enlace, el nivel de red enviará 4 paquetes de 64 bytes por segundo de extremo a extremo de la cadena. En la figura 2 comparamos el ratio de paquetes entregados por esta versión incompleta de LLRB frente a IEEE 802.11. Nuestro nivel de enlace aún no percibe si todos los paquetes han sido entregados (no hay estimación de vecindario), así que los resultados no los veremos directamente sino tal y como lo percibe el nivel de red. En el eje de abscisas se representa la longitud de la cadena, en el de ordenadas, el ratio de paquetes de red entregados.

Estas son unas condiciones muy poco exigentes, donde IEEE 802.11 entrega todos los paquetes enviados, sin duplicados. Contando sólo con el primer módulo de LLRB, el número de paquetes de red entregados es superior al de paquetes recibidos, esto es, hay paquetes de red duplicados. Estos duplicados se deben a pérdidas de asentimientos en el nivel de enlace: Todas las tramas de datos del nivel de enlace se entregan pero algunos ACK colisionan, no se reciben y el nivel de red reenvía el paquete. La detección de duplicados de IEEE 802.11 es suficiente para que ninguna trama llegue repetida, lo que se recibe es el mismo paquete de red en diferentes tramas de enlace.

El primer módulo de LLRB por sí mismo, sin contar con el resto del protocolo consigue un *throughput* de 16kbps (a partir del IEEE 802.11 de 2Mbps). La causa principal de esta disminución respecto a IEEE 802.11 es el aumento del espacio de tiempo antes de los CTS y de los ACK: pasa de ser SIFS a tomar el valor LIFS, que es bastante mayor. Los *timeouts* aumentan en la medida adecuada y por tanto el rendimiento disminuye respecto al protocolo original.

La razón de ser de LLRB es su uso en sistemas de recursos limitados, donde las técnicas convencionales *no caben*. Si las máquinas son de menores prestaciones sin duda el rendimiento será inferior, pero teniendo en cuenta que estamos aplicando técnicas nuevas donde las técnicas clásicas son inviables, podemos considerar que la mejora es $+\infty$. Además, LLRB es un protocolo para radiado, el rendimiento relevante será el visto por la suma de destinatarios, mientras que en esta configuración de prueba, desde el punto de vista del nivel de red hay un único destinatario de los datos.

6 CONCLUSIONES

En muchas ocasiones es deseable comunicar información con cierta fiabilidad a todas las estaciones adyacentes. Para esto, el primer paso es conocer cuáles son estas estaciones. Cuando las redes se forman y destruyen sobre la marcha y los recursos son limitados, a cada nodo le resulta costoso conocer la identidad de sus fugaces vecinos. 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*, y hasta donde sabemos, esto no está resuelto. En particular hemos detectado esta carencia en el protocolo de red ADSR, por ello desarrollamos el protocolo de enlace LLRB .

El primer paso de este radiado es extender el mecanismo RTS/CTS/ACK para más de un destinatario. Supone ampliar el concepto *portadora virtual*: Para el *unicast* fiable, el medio se *bloquea* con un RTS, y se *libera* con un CTS, que sólo un nodo concreto puede emitir. Para el *multicast* fiable, serán varios los nodos que envíen CTS. Además, el concepto *el medio está ocupado* ya no es absoluto: Podrá estar ocupado para enviar nuevas tramas de datos, pero tal vez no para tramas CTS o ACK.

La información que viaja desde un único origen a varios destinatarios no es conflictiva, la novedad es que las respuestas (CTS y ACK) deben compartir el medio, si bien tienen la ventaja de que el emisor, que es único, les puede coordinar. Si los recursos son limitados, este emisor no conoce a sus vecinos, pero puede hacer ciertas estimaciones.

En cuanto al trabajo futuro, estamos desarrollando los módulos segundo y tercero de LLRB. Otra tarea pendiente es pulir su integración con el protocolo de red para el que es diseñado, ADSR. El diseño actual no contempla que LLRB pueda convivir con 802.11 convencional, una tarea futura será permitir la coexistencia de ambos.

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 Commission. EU IST FET, Situated and Autonomic Communications (COMS) - Communication Paradigms for 2020. <http://www.cordis.lu/ist/fet/comms.htm>, July 2003.
- [4] K. Fall and K. Varadhan. The ns manual. <http://www.isi.edu/nsnam/ns/doc>. UC Berkeley and Xerox PARC.
- [5] 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.
- [6] P. Karn. MACA - a new channel access method for packet radio. In *Amateur Radio 9th Computer Networking Conference*, pages 134–140, 1990.
- [7] J. O. Kephart and D. M. Chess. The vision of autonomic computing. *IEEE Computer Magazine*, 36(1):41–50, 2003.
- [8] M. A. Ortuño, V. Matellán, L. Rodero, and J. Centeno. Abbreviated dynamic source routing: Protocolo DSR abreviado para máquinas con pocos recursos. In *Actas de las IV Jornadas de Ingeniería Telemática*, pages 385–391, 2003.
- [9] 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.
- [10] C. E. Perkins. *Ad Hoc Networking*. Addison-Wesley, 2001.
- [11] J. Weinmiller, H. Woesner, J. Ebert, and A. Wolisz. Analyzing the RTS/CTS mechanism in the DFWMAC media access protocol for wireless LAN's. In *IFIP TC6 Workshop Personal Wireless Communications*, 1995.