

Evaluación de mecanismos de priorización en 802.11p con VHDL

Juan Bautista Tomás Gabarrón, Esteban Egea López, Joan García Haro
Departamento de Tecnologías de la Información y las Comunicaciones,
Universidad Politécnica de Cartagena, Plaza del Hospital 1, 30202 Cartagena
{juanba.tomas, esteban.egea, joang.haro}@upct.es

Resumen—El estándar para comunicaciones inalámbricas en redes vehiculares, IEEE 802.11p, ofrecerá mecanismos de priorización de la información para permitir una diferenciación de la calidad de servicio entre aplicaciones. Debido a las condiciones especiales a las que están sujetas las redes vehiculares, es importante realizar una sintonización de los parámetros que configuran los procedimientos de diferenciación de servicios. En este artículo se presenta el desarrollo de un modelo en VHDL (*VHSIC HDL - Very High Speed Integrated Circuit Hardware Description Language*) que permite evaluar cómo reaccionan las interfaces de salida en un nodo que implementa IEEE 802.11p frente a distintas configuraciones de los parámetros de ajuste mencionados. El análisis de la distribución de ancho de banda entre interfaces permitirá discernir cómo se administra este reparto en nodos equipados con IEEE 802.11p.

Palabras Clave—Redes Vehiculares, FPGA, VANET, WAVE, 802.11p.

I. INTRODUCCIÓN

En los últimos diez años hemos asistido a un avance importante en el uso de redes inalámbricas para el acceso a Internet y para la formación de redes de ámbito local y personal (LANs y PANs) [1]. El estándar IEEE 802.11 [2] ha constituido desde sus inicios la solución más atractiva para la interconexión de terminales inalámbricos y así lo demuestra el gran número de dispositivos comercializados hasta la fecha que implementan esta tecnología [3]. A lo largo de todo este tiempo este estándar ha experimentado ciertas modificaciones que lo hacen más idóneo para las características especiales que reúne cada entorno de operación. Es el caso de IEEE 802.11a [2], que introduce modificaciones en la capa física mediante la utilización de *OFDM* (multiplexación por división de frecuencias ortogonales) para incrementar las tasas de transmisión, o algunas más recientes como 802.11e [2], que ofrece capacidades de calidad de servicio (QoS) mediante la diferenciación de servicios con interfaces de distinta prioridad.

En la actualidad están emergiendo nuevos mecanismos de aprovechamiento de los recursos que ofrecen las redes inalámbricas. Es el caso de las redes *ad-hoc* móviles (MANET). Este tipo de redes se caracteriza principalmente por estar constituido por nodos con capacidades de movilidad, lo que implica que los procesos de transmisión y recepción se vean afectados por el tiempo y la posición espacial de los mismos. Un caso específico de estas redes son las *VANET* (redes *ad-hoc* vehiculares), que intercambian información de interés o bien entre vehículos o entre vehículos e infraestructura, y proporcionan así a los pasajeros prestaciones avanzadas en materia de seguridad, control

del tráfico rodado, ocio y acceso a información. Con este propósito se ha desarrollado recientemente una modificación profunda del estándar de comunicaciones inalámbricas IEEE 802.11 [2] que pretende regular la transmisión de información en entornos vehiculares. Este estándar adopta de forma específica la denominación IEEE 802.11p [4]. Debido a las características particulares de las transmisiones en entornos de tráfico vehicular, el estándar implementa una serie de mecanismos que lo hacen más idóneo que el 802.11 original. Además, IEEE 802.11p hereda los procedimientos de diferenciación de servicios que ya contemplaba la extensión 802.11e mediante la creación de una serie de interfaces que permiten administrar el servicio de los paquetes según la prioridad que tengan asignada.

Es propósito de este artículo investigar sobre el mecanismo de priorización que IEEE 802.11p ofrece, mediante la implementación mediante lenguaje VHDL de las operaciones que el estándar detalla, para un dispositivo hardware programable *FPGA Xilinx Spartan 3E* [5]. El objetivo principal consiste en el análisis preliminar de la distribución de ancho de banda que la estructura de interfaces en un nodo equipado con 802.11p realiza, según los valores asignados a los parámetros que determinan la prioridad para cada interfaz de servicio. El resto del artículo se estructura de la siguiente forma: la sección 2 contempla una descripción detallada de los aspectos más importantes que describen la funcionalidad del estándar IEEE 802.11p para redes vehiculares; la sección 3 detalla algunos trabajos relacionados con el desarrollo VHDL para la evaluación de protocolos de comunicación inalámbrica; la sección 4 expone detalladamente el modelo generado con el lenguaje de especificación hardware VHDL [6]; en la sección 5 se exponen y discuten los resultados obtenidos. Finalmente en la sección 6 se concluye el estudio.

II. ARQUITECTURA WAVE

WAVE (Wireless Access in Vehicular Environments) constituye la arquitectura de protocolos que administra las capas de nivel de red, enlace, acceso al medio y física para las comunicaciones en VANETs [4] (ver Fig. 1). IEEE 802.11p [4] representa la propuesta realizada por el IEEE dentro de la arquitectura WAVE para la regulación de los mecanismos de acceso al medio en redes *ad-hoc* vehiculares (VANET). Durante la redacción de este artículo, IEEE 802.11p todavía se encuentra en fase de finalización, y se estima Abril de 2009 como posible fecha de publicación oficial del estándar por parte del *IEEE 802.11 Task Group p*.

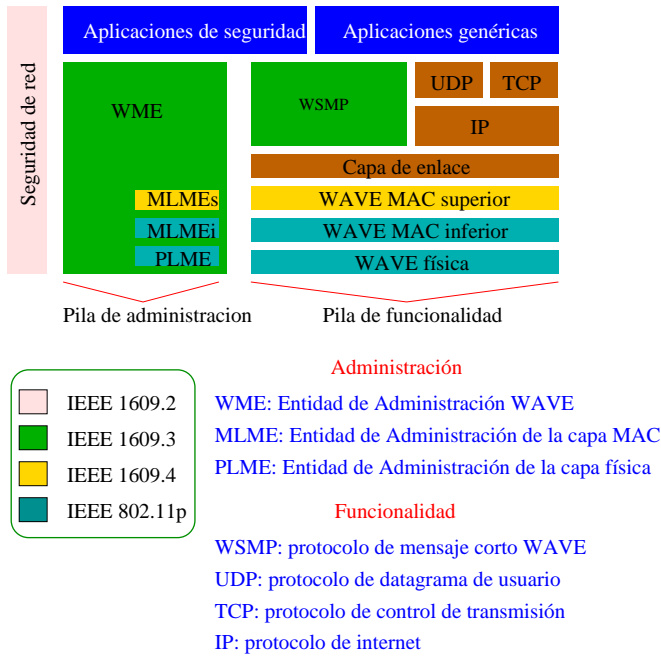


Fig. 1. Arquitectura de protocolos en WAVE (*Wireless Access in Vehicular Environments*)

A. DSRC (*Dedicated Short Range Communications*)

La historia de la arquitectura WAVE se remonta a 1999 cuando la FCC (*Federal Communications Commission*) estadounidense estableció un espectro de 75 MHz en la banda de los 5.9 GHz (banda de los ITS, *Intelligent Transportation Systems*) para albergar de manera exclusiva las tecnologías emergentes de radiocomunicaciones que tendrían lugar en las redes vehiculares de nueva generación. DSRC [7] (*Dedicated Short Range Communications*) es el nombre que adopta el espectro en esta banda, y se estructura según siete canales de 10 MHz cada uno (ver Fig. 2).

A diferencia de las comunicaciones en la banda de 2 GHz, cuya utilización está asociada a las redes WiFi y Bluetooth [8], la banda de los ITS requiere licencia para su uso.

El nivel físico propuesto por el DSRC se estructura en supertramas de duración predeterminada de 100 ms. Cada intervalo asociado a una supertrama se divide asimismo en dos períodos cada uno de los cuales se dedica a un aspecto particular de la comunicación (véase la Fig. 3). El primero de ellos es el CCH (canal de control), cuya duración por defecto es de 50 ms y que se encarga del envío de información importante, normalmente relacionada con la seguridad en la conducción, así como de la administración en la formación de grupos de entidades vehiculares según una aplicación determinada (subredes de ámbito específico). El segundo es el SCH (canal de servicio), compuesto por la multiplexación temporal cada 100 ms de canales que operan a distintas frecuencias dentro de la banda de los ITS. SCH permite la transmisión de información relacionada con seguridad, ocio y administración remota a través de paquetes IP (protocolo de Internet).

La diferencia fundamental entre CCH y SCH radica en el

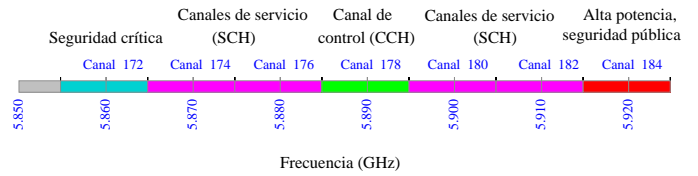


Fig. 2. Estructura del espectro en la banda de los ITS (*Intelligent Transportation Systems*)

hecho de que el primero no puede usar IP para la transmisión de paquetes. Para ello recurre a un protocolo de propósito específico que opera al mismo nivel que IP y que representa el acrónimo WSMP (*WAVE Short Message Protocol*) [4]. WSMP toma consideración de las características especiales que definen a los entornos de tránsito vehicular y reduce sustancialmente la carga de los paquetes (funcionalidad reducida de las cabeceras) para mejorar el caudal en las transmisiones.

B. IEEE 1609/802.11p

Los servicios de red de la arquitectura WAVE proporcionan capacidades de distribución de datos entre dispositivos WAVE y administración entre capas de la arquitectura de red. Esta última se compone de los estándares IEEE Std 1609.1TM-2006 (presentación), IEEE Std 1609.2TM-2006 (servicios de seguridad), IEEE Std 1609.3TM-2006 (servicios de red, WSMP), IEEE Std 1609.4-2006 y IEEE 802.11p (capas de acceso al medio). Por ser el eje central de nuestro desarrollo, nos centraremos en IEEE 802.11p, esquema fundamental de acceso al medio.

El esquema básico de acceso que contempla WAVE se denomina IEEE 802.11p y se basa en acceso múltiple por escucha de portadora (CSMA) [9]. Dicho mecanismo se denomina EDCA (*Enhanced Distributed Channel Access*) [4], y se utiliza tanto para el acceso en comunicaciones V2I (vehículo a infraestructura) como V2V (vehículo a vehículo), en esquemas de acceso cliente-servidor y distribuidos. Deriva de la DCF (Función de Coordinación Distribuida) especificada para el estándar original 802.11 [2], pero integra la diferenciación de servicios mediante la definición de cuatro interfaces virtuales de acceso que implementan funcionalidades de *backoff* para la transmisión de las tramas. Cada interfaz se configura con una serie de parámetros relativos al tamaño máximo de ventana de *backoff* así como intervalos de espaciado AIFS (*arbitrary inter-frame space*) que vendrán determinados por la clase de acceso que dicha interfaz tenga asignada. Esto permite establecer distintas calidades de servicio para los numerosos tipos de

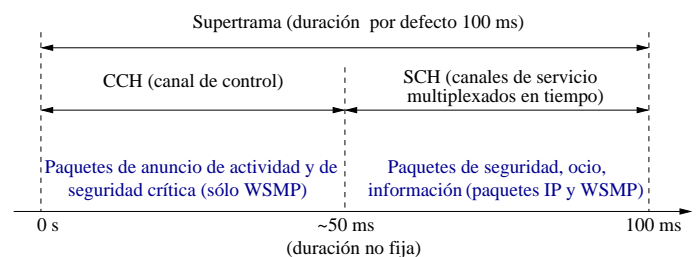


Fig. 3. Estructura particular de la supertrama en WAVE

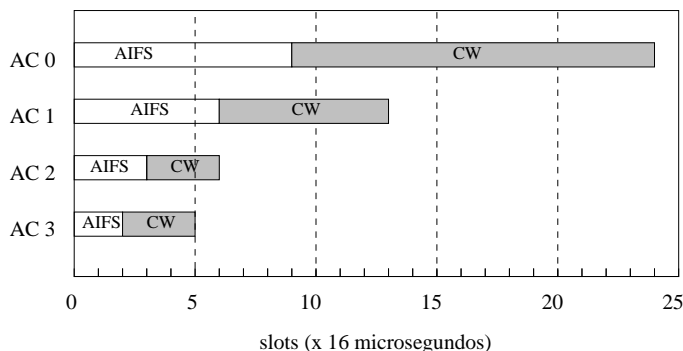


Fig. 4. Configuración de parámetros de *backoff* en IEEE 802.11p

aplicaciones que se prevé servir.

EDCA se basa en el servicio de paquetes de datos según la prioridad asignada a la aplicación que los genera. Para ello, cada estación mantiene configuraciones independientes para cada una de las cuatro interfaces definidas lo cual permite establecer distintas calidades de servicio (QoS) según el tipo de aplicación al que quiera darse servicio. Cada interfaz opera con mecanismos de acceso según escucha de portadora, es decir, cada estación virtual monitorizará el medio durante un intervalo determinado de tiempo (AIFS) a partir del cual ejecutará un proceso de *backoff* decremental en el que escogerá un número de slots aleatorio entre el tamaño máximo de ventana de *backoff* (CW_{max}) y el valor mínimo (CW_{min}) (ver Fig. 4). En caso de identificarse una colisión entre paquetes en el momento de transmitir, se reproducirá una nueva fase de *backoff* en la que los parámetros de anchura máxima y mínima de ventana se modificarán para reducir la probabilidad de colisión en intentos de acceso sucesivos (ver Tabla 1). Si se alcanza el número máximo de intentos posibles, se descarta el paquete y se reiniciarán los valores de CW a los iniciales para el nuevo paquete que se desee enviar al medio.

III. TRABAJOS RELACIONADOS

Algunos grupos de investigación se han interesado en la implementación física con VHDL de los protocolos asociados a redes inalámbricas basadas en 802.11. Todavía no existe ningún trabajo específico que se base en el estándar IEEE 802.11p, aunque sí existen numerosos estudios que evalúan los distintos estándares o modificaciones surgidos a partir del original (IEEE 802.11).

En primer lugar puede citarse la referencia [10], cuyos autores diseñaron la capa de acceso al medio (*enhanced MAC*) para la provisión de servicios multimedia sobre redes inalámbricas con 802.11. El prototipo diseñado soporta la funcionalidad de dos tipos de capa física: OFDM (*Orthogonal Frequency Domain Multiplexing*), basada en la utilización de frecuencias ortogonales con portadoras a 5 GHz para la modulación de la información a transmitir, y DSSS (*Direct Sequence Spread Spectrum*), que utiliza mecanismos de espectro expandido, con portadoras centradas a la frecuencia de 2 GHz. La implementación permite cambiar de un

mecanismo de capa física a otro de manera realmente sencilla.

En [11] se presenta el diseño de un prototipo basado en OFDM para la capa física que opera según el estándar 802.11a, muy similar en capa física a IEEE 802.11p. En dicho trabajo se realiza un estudio detallado de los mecanismos necesarios para la implementación de la capa física sobre una plataforma de diseño FPGA de Xilinx. En el sistema se incluye lógica de sincronización para la detección de paquetes y el establecimiento de los tiempos de operación. Se presenta además una metodología de trabajo realmente útil de cara al desarrollo de prototipos asociados a la comunicación en redes inalámbricas 802.11.

El estudio llevado a cabo en [12] introduce un prototipo desarrollado para implementar las comunicaciones a través del estándar 802.11a, aunque se centra en la construcción de una arquitectura *single chip* que permite el ahorro en costes de producción y en consumo de potencia. Para ello se pretende reducir la complejidad en el desarrollo y optimizar al máximo la integración de componentes.

En [13] se describe una arquitectura que implementa el protocolo 802.11a en una estructura *on-chip*. Se presenta el flujo completo de diseño del dispositivo, desde la simulación del protocolo mediante SDL (*Software Description Language*) [14] hasta la integración de componentes en la estructura *on-chip*. Asimismo se explotan los recursos de hardware dedicado para una mejora sustancial del rendimiento en tareas críticas.

IV. MODELO DEL SISTEMA

Como hemos podido observar, el funcionamiento del estándar de comunicaciones inalámbricas para redes vehiculares se basa en la configuración de una serie de parámetros que determinan la operación de cada interfaz virtual en los nodos equipados con 802.11p. De una manera más detallada, los parámetros que configuran la funcionalidad son los siguientes:

- Duración del intervalo arbitrario entre tramas (AIFS). Este valor es fijo para cada interfaz, y se debe completar siempre la duración de dicho intervalo antes de comenzar con la siguiente etapa de *backoff*.
- Ancho de la ventana de *backoff*
 - Tamaño máximo de la ventana (CW_{max}). Determina el máximo valor obtenible de hallar el número aleatorio previo a la etapa de *backoff* decremental.
 - Tamaño mínimo de la ventana (CW_{min}). Establece el valor mínimo seleccionable cuando se obtiene el valor aleatorio antes de comenzar el *backoff*.
- Longitud del intervalo de transmisión de paquetes para cada interfaz, o período de oportunidad para la transmisión (TXOP, *transmission opportunity*).

El objetivo que se persigue en este estudio es evaluar cómo reaccionan las interfaces frente a distintos valores de estos

tres parámetros. Para ello se define un modelo que emula la operación real de una estación equipada con 802.11p, y se efectúa una serie de pruebas en las que analizamos cómo opera el sistema en las situaciones que veremos en la siguiente sección.

La estructura generada con VHDL para el presente trabajo se basa en la operación de dos módulos principales donde cada uno de ellos desempeña una función particular pero estrechamente relacionada con la del otro. En primer lugar, existe un módulo denominado *backoffModule* el cual se encarga de llevar a cabo los procedimientos relacionados con las esperas en los intervalos de *backoff* para cada una de las cuatro interfaces que componen el sistema. Este módulo contiene asimismo una estructura generadora de números aleatorios que permite introducir la aleatoriedad requerida por el proceso de *backoff*. En segundo lugar, el módulo *control* tiene la función de controlar los procedimientos de *backoff* que se realizan en el primer módulo según las señales que éste provee. A continuación detallaremos la arquitectura generada en cada nivel de abstracción.

A. Estructura de primer nivel

La estructura de mayor nivel puede visualizarse en la Fig. 5. Esta implementación se realiza para facilitar las pruebas posteriores con la tecnología *FPGA Spartan 3E*. Como se puede observar, en esta estructura existe un total de siete entradas, cuya función se analiza a continuación:

- **reset.** Esta entrada está conectada directamente a un generador de números aleatorios que se encargará de dotar de aleatoriedad al procedimiento de *backoff*. Sólo se utiliza esta entrada para poner en funcionamiento el generador, ya que no es necesaria su reinicialización posterior una vez que se comienza con el proceso de generación.
- **enable.** Este puerto está conectado al generador de números aleatorios, y determina si se encuentra en funcionamiento o no.
- **aifs.** Este bus de entrada establece cuáles son los tiempos de espera entre tramas (AIFS) antes del procedimiento de *backoff*. Concretamente, es uno de los parámetros de configuración que usaremos para evaluar el comportamiento de la arquitectura. Para las cuatro interfaces existentes se destinan 3 bits que contendrán la información relativa al tamaño del período AIFS particular para cada una de ellas.
- **CWLength.** Este puerto de entrada de tipo bus proporciona la información relativa al tamaño máximo que puede tomar la ventana de contienda de cada interfaz. Éste es otro de los parámetros que modificaremos en las pruebas posteriores.
- **txop.** Este bus determina cuál es la duración máxima de cada interfaz para la transmisión de información al medio por cada intento de transmisión.
- **sload.** Puerto de entrada que sirve para hacer una pre-carga de los contadores al inicio de las simulaciones.
- **clk.** Señal de reloj del dispositivo.

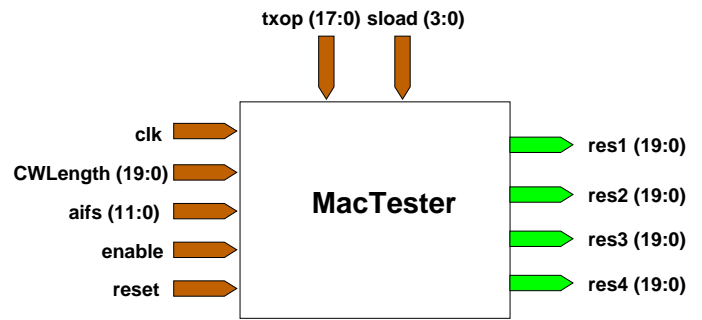


Fig. 5. Estructura de mayor nivel de la implementación en VHDL (en paréntesis se especifica el rango en bits de los puertos)

Las salidas de este dispositivo informan sobre cuál es el número de ciclos que se asigna a cada interfaz durante un intervalo determinado. En total son cuatro salidas, cada una de las cuales es un bus que proporcionará un valor relativo al tiempo que cada interfaz accede al medio. Cada salida recibe el nombre de *resX*, donde *X* ($X = 1..4$) se refiere al número de interfaz correspondiente, mientras que cada una posee un tamaño de 20 bits.

B. Estructuras de segundo nivel

En el segundo nivel (nivel de abstracción siguiente al anterior en orden descendente encontramos una estructura compuesta por *MaController* y *AccessCounter* (ver Fig. 6). El módulo más importante es el primero, que en definitiva guarda toda la lógica que implementa nuestro sistema. El segundo módulo es simplemente una estructura que contabiliza el número de ciclos que cualquiera de las cuatro salidas del bus del primer módulo permanece a 1 lógico (ciclo de tiempo asignado a una interfaz en particular).

C. Estructuras de tercer nivel

Esta estructura detalla los módulos que componen el dispositivo *MaController* descrito en el subapartado anterior. *MaController* implementa cuatro módulos de *backoff* cada uno correspondiente a sendas interfaces dentro de la arquitectura IEEE 802.11p (véase Fig. 7). Estos módulos están también conectados a un módulo de control que se encarga de administrar las señales necesarias para el procedimiento de *backoff*, como por ejemplo, la señal de ocupación del medio (*chBusy*) o la señal de aviso sobre fin de transmisión (*endTx*). A continuación detallamos estas entradas y otras que son en definitiva, las que caracterizan

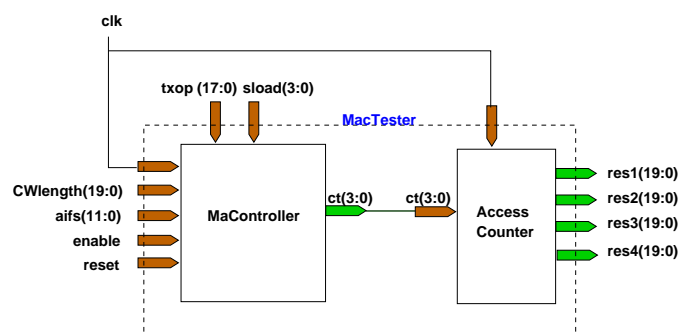


Fig. 6. Estructuras de segundo nivel de la implementación en VHDL (en paréntesis se especifica el rango en bits de los puertos)

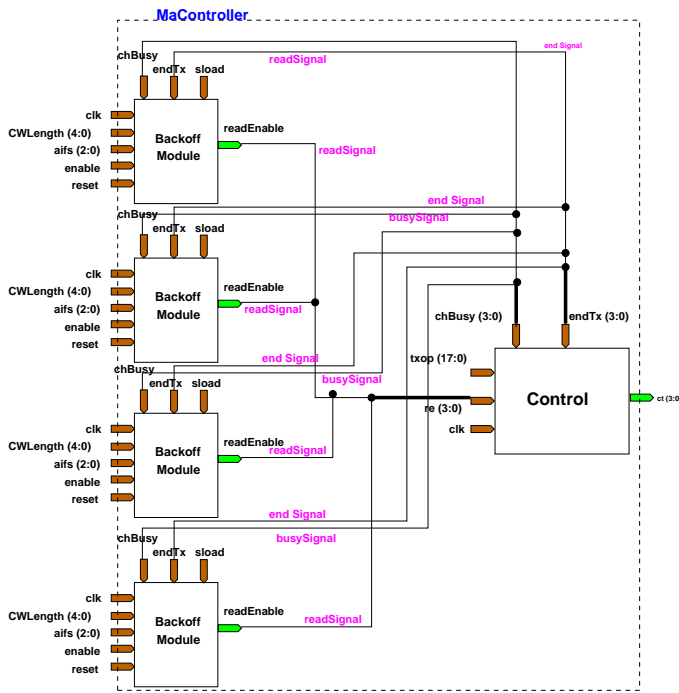


Fig. 7. Estructuras de tercer nivel de la implementación en VHDL (en paréntesis se especifica el rango en bits de los puertos)

este nivel de abstracción:

- **chBusy.** Esta entrada procede del elemento de control y tiene la misión de informar a cualquier módulo de *backoff* la situación de disponibilidad del canal: libre (0) u ocupado (1). En caso de que esté ocupado, el módulo de *backoff*, o bien esperará para iniciar el proceso de *backoff* hasta que quede nuevamente libre, o bien si ya había comenzado uno lo posterga hasta el siguiente momento en el que se pueda monitorizar el canal como desocupado.
- **readEnable.** Ésta es la única salida de cada uno de los módulos de *backoff*, y determina en un momento en particular si ha terminado el proceso de *backoff* para una interfaz concreta (1) o todavía se encuentra en estado de decremento (0). La señal de cada módulo de *backoff* irá a parar al módulo de *Control*, para que sea éste quien administre qué estación virtual tiene el derecho a transmitir en un momento determinado según el esquema de prioridades que se haya asignado previamente.
- **endTx.** Esta señal indica a aquella interfaz que haya comenzado previamente una transmisión, si dicha emisión de información ha terminado ya (mediante un pulso). Se utilizará para administrar las pausas de los módulos de *backoff* y las señales que gobiernan el estado actual del medio (libre u ocupado).
- **txop.** En el momento de capturar el medio para transmisión, cada estación tendrá derecho a emitir paquetes al medio durante un intervalo de tiempo determinado. Este intervalo de tiempo viene dado por los parámetros de configuración que se hayan asignado a la estructura por medio de este puerto de entrada.

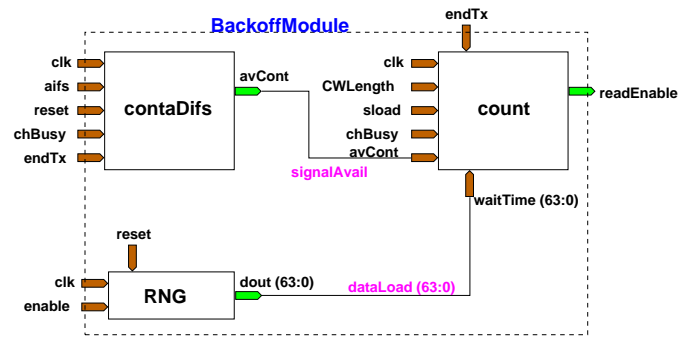


Fig. 8. Estructuras de cuarto nivel de la implementación en VHDL (en paréntesis se especifica el rango en bits de los puertos)

D. Estructuras de cuarto nivel

Los módulos que todavía permiten ahondar más en su implementación interna son específicamente los de *backoff*. Concretamente se basan en la utilización de dos contadores y un generador de números pseudoaleatorios (véase Fig. 8). En primer lugar, un contador realiza la cuenta atrás correspondiente al período AIFS que cada interfaz en particular debe realizar cuando hay información para transmitir y además se monitoriza el medio libre. Una vez se determina que el medio ha quedado libre tras una transmisión, el contador AIFS informa al contador de *backoff* que ya puede comenzar el procedimiento decremental del mismo nombre. Al mismo tiempo, cuando comienza el período AIFS, el generador de números pseudoaleatorios proporciona un valor de 32 bits que se usa para inicializar el contador de *backoff* a un número de slots aleatorio (dentro de los márgenes impuestos por el valor procedente del puerto AIFS) que se usará en el momento en el que se comience con el procedimiento.

V. EXPERIMENTACIÓN

En esta sección se incluye una serie de pruebas experimentales con variaciones en los valores propios de los parámetros de *backoff* para comprobar cómo se establece la prioridad entre las distintas interfaces que componen el sistema. Todas las pruebas se realizarán a un tiempo de funcionamiento de 10 ms, intervalo más que suficiente para asegurar la operación del sistema en régimen permanente, determinado por la utilización de una frecuencia de reloj de 100 ns que permite efectuar los experimentos con aproximadamente 100000 ciclos de reloj totales. Debido a la pequeña cantidad de ciclos de reloj asociados a los parámetros AIFS y TXOPs, que determinan los tiempos de ocupación, frente al número total de ciclos mencionados en emulación, es obvio el alcance del régimen permanente.

Cada subsección de esta parte se corresponderá con una configuración particular de parámetros, se analizarán los resultados obtenidos en la gráfica de la Figura 9 y se terminará con una justificación de los mismos. Concretamente, en dicha Figura se represetan los resultados de cada experimento (marcados con un color distinto para cada uno de los experimentos) obtenidos para cada una de las interfaces; en el eje de abscisas cada interfaz se identifica con un número de 1 a 4. Estos resultados se corresponden con el porcentaje de ocupación de cada interfaz respecto al total (número total de ciclos

Tabla I
VALORES DE CONFIGURACIÓN PARA CADA EXPERIMENTO

	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
<i>CWMax1</i>	7 ciclos	7 ciclos	7 ciclos	7 ciclos	7 ciclos
<i>CWMax2</i>	7 ciclos	7 ciclos	5 ciclos	7 ciclos	5 ciclos
<i>CWMax3</i>	7 ciclos	7 ciclos	4 ciclos	7 ciclos	4 ciclos
<i>CWMax4</i>	7 ciclos	7 ciclos	3 ciclos	7 ciclos	3 ciclos
<i>txop1</i>	7 ciclos	7 ciclos	7 ciclos	3 ciclos	3 ciclos
<i>txop2</i>	7 ciclos	7 ciclos	7 ciclos	7 ciclos	7 ciclos
<i>txop3</i>	7 ciclos	7 ciclos	7 ciclos	15 ciclos	15 ciclos
<i>txop4</i>	7 ciclos	7 ciclos	7 ciclos	31 ciclos	31 ciclos
<i>AIFS1</i>	3 ciclos	5 ciclos	3 ciclos	3 ciclos	2 ciclos
<i>AIFS2</i>	3 ciclos	4 ciclos	3 ciclos	3 ciclos	3 ciclos
<i>AIFS3</i>	3 ciclos	3 ciclos	3 ciclos	3 ciclos	4 ciclos
<i>AIFS4</i>	3 ciclos	2 ciclos	3 ciclos	3 ciclos	5 ciclos

correspondientes a 10 ms). El carácter del tráfico utilizado se corresponde con el de una fuente de tipo *greedy* (siempre existe información para transmitir).

A. Experimento 1

Esta primera prueba tiene el objeto de observar cómo reaccionan las interfaces cuando el valor de los parámetros de *backoff* para cada interfaz es idéntico, es decir, tanto los anchos de ventana, como las duraciones de los intervalos AIFS y TXOPs serán iguales para cada clase de prioridad. El resultado debería ser unos tiempos de acceso al medio estadísticamente muy similares para cada clase. Si observamos la configuración establecida para este caso (ver Tabla I, columna Exp. 1) obtenemos unos resultados como los mostrados en la Fig. 9 para la tonalidad negra. Se puede observar que la distribución de ancho de banda es equitativa, con la salvedad de que aquella clase de acceso de mayor prioridad obtendrá una tasa ligeramente mayor de accesos que el resto por el hecho de que en aquellos casos en los que se produzcan colisiones entre interfaces dentro de la misma estación, la de mayor prioridad (en este caso, la 4) será quien finalmente obtenga el derecho para transmitir.

B. Experimento 2

En esta nueva prueba se mantienen fijos los valores asociados a la anchura de las ventanas y a las oportunidades de transmisión, como puede observarse en la Tabla I para la columna Exp. 2. Los únicos parámetros que se diferencian se corresponden con los espaciados de tiempo entre tramas AIFS). Según los resultados correspondientes (Fig. 9, representados por la tonalidad amarilla), es posible ver de manera muy evidente que aquellas interfaces con menores tiempos de AIFS obtienen un mayor número de accesos al medio, lo que prueba la distinción entre clases de prioridad. Además, en este caso se obtiene una tasa de utilización del canal algo menor que la del caso anterior, sobre todo debido a la limitación de las clases de menor prioridad a las que se les asigna menores anchos de banda.

C. Experimento 3

En este caso se modifican los anchos máximos de ventana admisibles para cada interfaz, manteniéndose el orden de

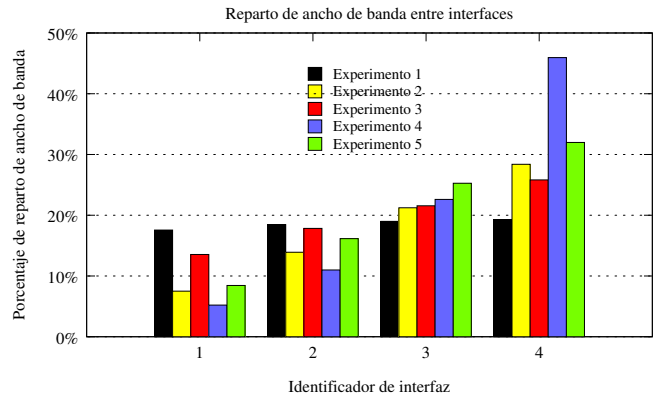


Fig. 9. Relación de ancho de banda asignado para cada interfaz en cada experimento

la asignación de prioridades como en el caso anterior (ver Tabla I, columna Exp. 3). También pueden observarse en este caso las distintas prestaciones en cuanto a la adquisición del medio para cada clase de prioridad según el máximo ancho de ventana de *backoff* (ver Fig. 9, tonalidad roja). Sin embargo, es necesario resaltar que en este caso la distinción entre porcentajes de distribución de ancho de banda no es tan alta entre interfaces como en el Experimento 2 debido al hecho de que en el procedimiento de *backoff* se obtienen números aleatorios entre 0 y el tamaño máximo de la ventana (*CWMax*), mientras que en la Prueba anterior siempre debía contabilizarse el número total de ciclos correspondientes al AIFS de la clase que quiere establecer la transmisión. Como consecuencia, no se produce la misma "injusticia" de reparto del ancho de banda entre las interfaces de menor prioridad como en el caso previo, y por este motivo la utilización del medio también se incrementa considerablemente, rondando el 80% (resultado de sumar los porcentajes asociados a cada interfaz).

D. Experimento 4

Según la configuración expuesta para este caso (ver Tabla I, columna Exp. 4), en el que se varían los tiempos de adquisición del medio para cada clase de prioridad, en esta prueba deberían obtenerse los tiempos de adquisición del medio más diferentes para cada interfaz, ya que la magnitud de la configuración de cada *txop* es también mayor. Puede observarse en la Fig. 9, tonalidad morada, que en comparación con los casos anteriores, para mayores valores de *txop* se obtienen diferencias aun más notables en la distribución de los accesos según la clase de prioridad. Asimismo, a pesar del limitado ancho de banda asignado a las clases de menor prioridad, en este caso como es lógico la utilización del medio es mayor porque se reducen los tiempos dedicados al procedimiento de *backoff* y AIFS, con una utilización de casi el 85%.

E. Experimento 5

En este caso se realiza una configuración de parámetros que engloba todos los cambios efectuados en las pruebas anteriores (ver Tabla I, columna Exp. 5). En la Fig. 9, tonalidad verde, podemos observar que, aunque los AIFS están configurados para dar más prioridad a las clases con orden decreciente (AIFS1 tiene más prioridad que AIFS2 y así

sucesivamente), el parámetro que más influencia tiene sobre el resultado final es precisamente el correspondiente a las duraciones de tiempo asignadas a cada clase de acceso en particular (*txop*). Por ello puede inferirse que *txop* determina en gran medida cuál va a ser la asignación de ancho de banda a cada entidad, lo que sin embargo supondrá una mayor limitación de ancho de banda en las clases de menor prioridad, pero a la vez la utilización del medio también será más grande.

VI. CONCLUSIONES Y TRABAJO FUTURO

En este artículo se describe la implementación de un único esquema de interfaces bajo 802.11p en VHDL. Dicho estudio pretende ser un trabajo preliminar para el diseño de un dispositivo sobre hardware reconfigurable que permita el establecimiento de comunicaciones entre dispositivos equipados con el estándar IEEE 1609/802.11p (WAVE). Las pruebas a las que ha sido sometida la arquitectura diseñada se basan en analizar el comportamiento entre clases de servicio para evaluar la dependencia de las distribuciones de ancho de banda respecto de la configuración de parámetros. Debido a que la pretensión de este primer experimento era simplemente evaluar las prestaciones internas de un dispositivo aislado, sin entrar en el análisis de la funcionalidad referida a multitud de equipos en comunicación simultánea, los resultados, aunque prometedores, se refieren a condiciones de mejor caso. Esto resta realismo a los experimentos, ya que en una situación real se dispondría de una gran cantidad de nodos en comunicación simultánea que, por otra parte, supondría una degradación notable en la comunicación y asignaciones de ancho de banda por interfaz bastante menores a las que hemos relatado. Sin embargo, en este primer trabajo se ha podido entrever algunos aspectos clave para el funcionamiento del protocolo que ayudan a comprender su funcionalidad de manera detallada.

Por otra parte, a partir de los experimentos se ha podido poner de relieve la enorme influencia que tienen los parámetros de configuración sobre las prestaciones de diferenciación de servicios ofrecidas por las interfaces de los dispositivos 802.11p. Se ha determinado con claridad que los parámetros relativos a las longitudes de ventana, intervalos de tiempo entre tramas y los períodos de oportunidad para transmitir ofrecen muchas posibilidades de cara a optimizar la funcionalidad de la arquitectura de interfaces. A modo de ejemplo, se podría establecer una configuración dinámica de estos parámetros en función de la información provista por las capas de aplicación y del entorno de operación, para dar mayor prioridad a una clase respecto a otra bajo circunstancias especiales de la comunicación (si requiere tiempo real, si es orientada a conexión, etc.). En la actualidad el borrador del estándar IEEE 802.11p recomienda unos valores predeterminados para estos parámetros de configuración, aunque en el documento oficial se sugiere modificar sus valores para adaptar el funcionamiento al entorno de operación concreto.

A modo de conclusión, el nivel de implementación que se ha alcanzado con la presente arquitectura en VHDL se puede extender bastante más. En cuanto al trabajo futuro, con carácter inmediato se trabaja en la obtención de los retardos

asociados a los tiempos de acceso al medio para una versión mejorada de la implementación, que permite configurar el sistema con parámetros más realistas (incluidos los referentes al estándar IEEE 802.11p), y al mismo tiempo permite la integración de la funcionalidad de colas de paquetes, lo que incrementa notablemente el realismo de los resultados. Además se pretende continuar con el desarrollo de un dispositivo completo que implemente la totalidad de los protocolos de la arquitectura WAVE y permita realizar en un futuro mediciones de campo que aporten mejoras cuantitativas respecto de la simulación convencional.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto nacional TEC2007-67966-01/TCM (CON-PARTE-1), el proyecto Regional 00002/CS/08 de la Fundación Séneca y está también enmarcado en el Programa de Ayudas a Grupos de Excelencia de la Región de Murcia, Fundación Séneca”.

REFERENCIAS

- [1] Hua Zhu, Ming Li, I. Chlamtac, and B. Prabhakaran. A survey of quality of service in IEEE 802.11 networks. *Wireless Communications, IEEE*, 11(4):6–14, Aug. 2004.
- [2] IEEE-SA Standards Boards, editor. *IEEE Standard for Information technology, Telecommunications and information exchange between systems, Local and metropolitan area networks, Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Computer Society, 2007.
- [3] Frost and Sullivan Consulting. Wireless industry statistics. Disponible en <http://www.callcentermagazine.com/article/COM20010822S0007>. Acceso en Marzo 2009.
- [4] IEEE-SA Standards Boards, editor. *IEEE 1609/802.11p - Family of Standards for Wireless Access in Vehicular Environments (WAVE)*. IEEE Computer Society, 2007.
- [5] Xilinx Corporation. Spartan-3e starter kit. Disponible en www.xilinx.com/products/devkits/HW-SPAR3E-SK-US-G.htm. Acceso en Marzo 2009.
- [6] EDA Industry Working Groups. Vhdl international (vi). Disponible en <http://www.vhdl.org/>. Acceso en Abril 2009.
- [7] ABIresearch. Dedicated short-range communications (dsrc). Disponible en [http://www.abiresearch.com/research/1000690-Dedicated_Short_Range_Communications_\(DSRC\)](http://www.abiresearch.com/research/1000690-Dedicated_Short_Range_Communications_(DSRC)). Acceso en Marzo 2009.
- [8] E. Ferro and F. Potorti. Bluetooth and wi-fi wireless protocols: a survey and a comparison. *Wireless Communications, IEEE*, 12(1):12–26, Feb. 2005.
- [9] Tien-Shin Ho and Kwang-Cheng Chen. Performance analysis of IEEE 802.11 CSMA/CA medium access control protocol. volume 2, pages 407–411 vol.2, Oct 1996.
- [10] Y. Kim, H. Jung, H. H. Lee, Electron. Cho K. R. Router Technol. Dept., and Taejon; Telecommun. Res. Inst. Mac implementation for IEEE 802.11 wireless lan. *Personal Communications, IEEE*, pages 191–195, Apr 2001.
- [11] F. Manavi and Y.R. Shayan. Implementation of OFDM modem for the physical layer of IEEE 802.11a standard based on Xilinx Virtex-II FPGA. *Personal Communications, IEEE, Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, 3(6):1768–1772, May 2004.
- [12] E. Grass, K. Tittelbach-Helmrich, U. Jagdhold, A. Troya, G. Lippert, O. Kruger, J. Lehmann, K. Maharatna, K.F. Dombrowski, N. Fiebig, R. Kraemer, and P. Mahonen. On the single-chip implementation of a HyperLAN/2 and IEEE 802.11a capable modem. *Personal Communications, IEEE*, 8(6):48–57, Dec 2001.
- [13] G. Panic, D. Dietterle, Z. Stamenkovic, and K. Tittelbach-Helmrich. A system-on-chip implementation of the IEEE 802.11a MAC layer. pages 319–324, Sept. 2003.
- [14] Rachida Dssouli, Gregor von Bochmann, and Yair Lahav, editors. *SDL '99 The Next Millennium, 9th International SDL Forum, Montréal, Québec, Canada, 21-25 June, 1999, Proceedings*. Elsevier, 1999.